Hindawi Publishing Corporation
The Scientific World Journal
Volume 2013, Article ID 969734, 12 pages
http://dx.doi.org/10.1155/2013/969734

Research Article

An Adaptive Cauchy Differential Evolution Algorithm for Global Numerical Optimization

Tae Jong Choi,[1] Chang Wook Ahn,[1] and Jinung An[2]

[1] *Department of Computer Engineering, Sungkyunkwan University (SKKU), 2066 Seobu-ro, Suwon 440-746, Republic of Korea*
[2] *Robot Research Division, Daegu Gyeongbuk Institute of Science and Technology (DGIST), 50-1 Sang-ri, Hyeonpung-meyeon, Daegu 711-873, Republic of Korea*

Correspondence should be addressed to Chang Wook Ahn; cwan@skku.edu

Received 3 May 2013; Accepted 6 June 2013

Academic Editors: P. Agarwal, S. Balochian, V. Bhatnagar, J. Yan, and Y. Zhang

Adaptation of control parameters, such as scaling factor (F), crossover rate (CR), and population size (NP), appropriately is one of the major problems of Differential Evolution (DE) literature. Well-designed adaptive or self-adaptive parameter control method can highly improve the performance of DE. Although there are many suggestions for adapting the control parameters, it is still a challenging task to properly adapt the control parameters for problem. In this paper, we present an adaptive parameter control DE algorithm. In the proposed algorithm, each individual has its own control parameters. The control parameters of each individual are adapted based on the average parameter value of successfully evolved individuals' parameter values by using the Cauchy distribution. Through this, the control parameters of each individual are assigned either near the average parameter value or far from that of the average parameter value which might be better parameter value for next generation. The experimental results show that the proposed algorithm is more robust than the standard DE algorithm and several state-of-the-art adaptive DE algorithms in solving various unimodal and multimodal problems.

1. Introduction

Differential Evolution (DE) is a powerful population based search technique for optimizing problems. Many researchers have used the DE in practical fields because this technique has good convergence properties and is easy to apply [1]. The DE has three control parameters such as scaling factor (F), crossover rate (CR), and population size (NP). The performance of DE is largely influenced by what parameter values are assigned to these control parameters. Therefore, in order to have a good optimization performance, finding suitable parameter values is of crucial importance [2, 3]. In the DE, the control parameters are usually adjusted by using the trial-and-error search method. However, the value assigned by the trial-and-error method might be efficient for solving one type of problems and inefficient for solving other problems [4]. Moreover, it requires a lot of computational resources. As a solution of this problem, parameter adaptation has been utilized. According to Eiben et al. [5, 6], the parameter adaptation can be categorized into three classes as follows.

(1) Deterministic parameter control: the control parameters are adapted by some deterministic rule.

(2) Adaptive parameter control: the control parameters are adapted by some form of feedback from evolutionary search.

(3) Self-adaptive parameter control: the control parameters are adapted by the evolution-of-evolution technique. The control parameters are encapsulated in each individual as additional chromosomes and undergo evolutionary procedure.

The well designed adaptive or self-adaptive parameter control method can improve the performance of DE. Therefore, the adaptive and self-adaptive parameter controls are more applicable than the trial-and-error search method. So far, many adaptive and self-adaptive DE algorithms have been proposed and they have shown that the adaptive and self-adaptive DE algorithms have more robust performance than standard DE algorithm for many benchmark functions

[4, 7, 8]. Although there are many suggestions for adapting control parameters, it is still a challenging task to properly adapt the control parameters for problem. Based on various experiments, we found out that the parameter adaptation should be performed in every generation and the control parameters of each individual should be adapted based on the average parameter value of successfully evolved individuals' parameter values by using the Cauchy distribution.

The Cauchy distribution is one of the long tail distributions. The Cauchy distribution generates large step from peak location with higher probability. Many evolutionary algorithms have used this long tail property as an escaping local minima method. The proposed algorithm also, but in different manner, utilizes the Cauchy distribution for the parameter adaptation. In the proposed algorithm, each individual has its own control parameters. The control parameters of each individual are adapted based on the average parameter value of successfully evolved individuals' parameter values by using the Cauchy distribution. It is because the successfully evolved individuals are led by appropriate parameter values. That is to say, the appropriate parameter values make the individuals take the good region for solving problem. However, there is a possibility that the current appropriate parameter values might be inappropriate parameter values in next generation. Therefore, we cannot assure that the parameter adaptation based on the average parameter value is good for making well-suited parameter values for future generations. In view of the above considerations, the parameter adaptation of proposed algorithm utilizes the Cauchy distribution as a large step method. According to it, the control parameters of each individual are assigned either near the average parameter value or far from that of the average parameter value which might be better parameter value for next generation. The experimental results show that the proposed algorithm is more robust than standard DE algorithm and some adaptive and self-adaptive DE algorithms such as jDE [4], SaDE [7], and MDE [9] on solving multimodal problems as well as unimodal problems.

The rest of this paper proceeds as follows. In Section 2, we introduce basic operations of standard DE algorithm and some adaptive and self-adaptive parameter control DE algorithms. In Section 3, the Cauchy distribution is described. The proposed algorithm is explained in detail in Section 4. Section 5 presents the experimental results. We conclude this paper in Section 6.

2. Related Work

2.1. DE Algorithm. In the DE, a parent vector is called "target vector," a mutant vector is that generated by mixing donor vectors, and an offspring obtained by making crossover between target vector and mutant vector is called "trial vector." A target vector generates a trial vector which is moved around in search space by using the mutation and the crossover operations. If the fitness value of the trial vector is better than or equal to the fitness value of the target vector, the trial vector is accepted and included in the population of next generation, otherwise it is discarded and the target vector remains for the next generation. This cycle of operations is repeated until some specific termination conditions are not satisfied.

2.1.1. Initialization. The population of the DE consists of NP individuals. Each individual is a D-dimensional parameter vectors, denoted as $X_{i,G} = x^1_{i,G}, \ldots, x^D_{i,G}$ where $i = 1, \ldots,$ NP. In the initialization stage, first of all, the DE designates the search space of the test problem by prescribing the minimum ($X_{\min} = x^1_{\min}, \ldots, x^D_{\min}$) and the maximum ($X_{\max} = x^1_{\max}, \ldots, x^D_{\max}$) parameter bounds. After that, the parameter vectors of the each individual are initialized as follows:

$$x_{i,j,0} = x_{j,\mathrm{MIN}} + \mathrm{rand}\,[0,1] \cdot \left(x_{j,\mathrm{MAX}} - x_{j,\mathrm{MIN}}\right), \quad (1)$$

where rand[0, 1] is the uniform distributed random number lying between 0 and 1. By doing this, all the individuals are randomly scattered in the search space. After initialization, the DE executes a loop of the operations: mutation, crossover, and selection.

2.1.2. Mutation Operation. The mutation is the first operation to generate child individuals from their parent individuals. So far, a lot of mutation strategies have been proposed. Here, we explain an example, called DE/rand/1/bin, which was introduced by Storn and Price [1]. First of all, the mutation strategy randomly select three mutually exclusive individuals among [1, NP]. They are called the "donor vectors", denoted as $X_{r_1,G}$, $X_{r_2,G}$, and $X_{r_3,G}$. A mutant vector $V_{i,G}$ is generated by adding the scaling difference of $X_{r_2,G}$ and $X_{r_3,G}$ to $X_{r_1,G}$.

$$V_{i,G} = X_{r_1,G} + F \cdot \left(X_{r_2,G} - X_{r_3,G}\right), \quad (2)$$

where F is a scaling factor for amplifying the difference value between $X_{r_2,G}$ and $X_{r_3,G}$.

2.1.3. Crossover Operation. The crossover generates the trial vectors by making a crossover between target vector and mutant vector. There are two crossover operations which are commonly used. They are the binomial and the exponential crossovers. Here, we describe the binomial crossover. At the beginning, a random number is selected. If the random number is less than or equal to the crossover rate CR, the first element of the trial vector is occupied by the first element of the mutant vector. Otherwise, the element is occupied by the target vector. This procedure is repeated D times for each individual. The trial vector $U_{i,G} = u^1_{i,G}, \ldots, u^D_{i,G}$ is generated as follows:

$$u_{i,j,G} = \begin{cases} v_{i,j,G} & \text{if } (\mathrm{rand}\,[0,1) \le \mathrm{CR} \text{ or } j = j_{\mathrm{rand}}) \\ x_{i,j,G} & \text{otherwise.} \end{cases} \quad (3)$$

Prior to crossover, the DE select another random number j_{rand} lying between $[1, D]$. The random number is used to guarantee that at least one element of the trial vector is occupied by the mutant vector.

2.1.4. Selection Operation. The selection is the last operation of the DE iterations. It compares the fitness value of the target and the trial vectors. If the fitness value of the trial vector is better than or equal to the fitness value of the target vector, the trial vector is accepted and forms part of the population, otherwise it is discarded and the target vector remains for the next generation. these procedures are formulated as follows:

$$X_{i,G+1} = \begin{cases} U_{i,G} & \text{if } \left(f\left(U_{i,G}\right) \le f\left(X_{i,G}\right)\right), \\ X_{i,G} & \text{otherwise.} \end{cases} \quad (4)$$

2.2. jDE. Brest et al. [4] proposed a self-adaptive parameter control DE (called jDE) based on DE/rand/1/bin. In jDE, the control parameters, F and CR, are encapsulated in each individual as additional chromosomes. Therefore, all individuals have their own control parameters, denoted as F_i and CR_i. jDE utilizes four additional parameters: τ_1, τ_2, F_l, and F_u. The first two parameters are used to determine whether the control parameters need to be updated or not and the last two parameters are used to designate the range of the control parameter F_i. At the beginning, the values of F_i and CR_i are initialized by 0.5 and 0.9, respectively. Then, the control parameters F_i and CR_i for next generation are adapted as follows:

$$F_{i,G+1} = \begin{cases} F_l + \mathrm{rand}_2\left[0,1\right] \cdot F_u & \text{if } \left(\mathrm{rand}_1\left[0,1\right] < \tau_1\right), \\ F_{i,G} & \text{otherwise,} \end{cases}$$
$$\mathrm{CR}_{i,G+1} = \begin{cases} \mathrm{rand}_4\left[0,1\right] & \text{if } \left(\mathrm{rand}_3\left[0,1\right] < \tau_2\right), \\ \mathrm{CR}_{i,G} & \text{otherwise.} \end{cases} \quad (5)$$

The author used $\tau_1 = 0.1$, $\tau_2 = 0.1$, $F_l = 0.1$, and $F_u = 0.9$. This procedure is executed before applying the mutation operation. Therefore, newly generated control parameters affect the mutation and the crossover operations.

2.3. DESAP. Teo [10] proposed the first self-adaptive population size DE (called DESAP) based on the self-adaptive Pareto DE [11]. DESAP can self-adapt not only the scaling factor F and the crossover rate CR but also the population size NP. This algorithm utilizes additional parameters such as η_i, δ_i, and π_i. These parameters are encapsulated in each individual as additional chromosomes and also participated in the mutation and the crossover operations for evolving itself. The newly generated control parameters are selected when the fitness value of the trial vector is lower than or equal to the fitness value of the target vector. DESAP is divided into two algorithms (i.e., DESAP-abs and DESAP-rel) according to the equation of the population size for the next generation. DESAP has shown the effectiveness of the self-adaptive population size technique.

2.4. JADE. Zhang and Sanderson [12, 13] proposed a new mutation strategy called DE/current-to-$p$best which is lower greedy than DE/current-to-best/1. This strategy utilizes not the best individual of the population but the randomly selected one from the top $100p\%$ ($p \in (0,1]$) individuals. In addition, an external archive scheme was proposed by storing the set of parameter vectors of recently discarded individuals. These parameter vectors provide the additional information about promising progress direction and increase the population diversity. The following equations represent the DE/current-to-$p$best with and without the external archive strategy:

(1) DE/current-to-$p$best with archive:

$$V_{i,G} = X_{i,G} + F_i \cdot \left(X^p_{\mathrm{best},G} - X_{i,G}\right) + F_i \cdot \left(X_{r1,G} - \widetilde{X}_{r2,G}\right), \quad (6)$$

(2) DE/current-to-$p$best without archive:

$$V_{i,G} = X_{i,G} + F_i \cdot \left(X^p_{\mathrm{best},G} - X_{i,G}\right) + F_i \cdot \left(X_{r1,G} - X_{r2,G}\right), \quad (7)$$

where $\widetilde{X}_{r2,G}$ is an individual randomly selected from the population or external archive.

In terms of parameter adaptation, JADE adapts the crossover rate CR_i, as follows:

$$\mathrm{CR}_i = \mathrm{rnd}\, n_i\left(\mu_{\mathrm{CR}}, 0.1\right), \quad (8)$$

where $\mathrm{rnd}\, n_i$ is the Gaussian distributed random number generator. After that, the crossover rate CR_i is truncated to $[0,1]$. Moreover, μ_{CR} that is a mean value to generate CR_i is modified as follows:

$$\mu_{\mathrm{CR}} = (1-c) \cdot \mu_{\mathrm{CR}} + c \cdot \mathrm{mean}_A\left(S_{\mathrm{CR}}\right), \quad (9)$$

where c is a constant value in $[0,1]$, mean_A stands for the arithmetic mean, and S_{CR} contains the successfully evolved crossover rates of individuals after the selection operation. Similarly, the scaling factor F_i is adapted as follows:

$$F_i = \mathrm{rnd}\, c_i\left(\mu_F, 0.1\right), \quad (10)$$

where $\mathrm{rnd}\, c_i$ is the Cauchy distributed random number generator. After that, the scaling factor F_i is truncated to 1 if $F_i \ge 1$ or regenerated if $F_i \le 0$. Also, μ_F that is a mean value to generate F_i is modified as follows:

$$\mu_F = (1-c) \cdot \mu_F + c \cdot \mathrm{mean}_L\left(S_F\right), \quad (11)$$

where c is a constant value in $[0,1]$, mean_L stands for the Lehmer mean, and S_F contains the successfully evolved scaling factors of individuals after the selection operation.

2.5. MDE. Ali and Pant [9] proposed a Modified Differential Evolution (MDE). This algorithm utilizes the Cauchy distribution as another mutation operation. In the selection operation, all individuals are monitored and the results of the selection operation are stored in the failure counter. If some individuals consequently fail to be selected as an individual for the next generation over MFC (Maximum Failure Counter), MDE assumes that these individuals were felled into some local minima. Therefore, the algorithm applies the Cauchy distributed mutation to these individuals instead of the mutation and the crossover operations to escape the local minima. After that, the failure counter is initialized by 0. MDE has shown the good performance for the higher dimensional problems, compared with DE/rand/1/bin.

3. Analysis of the Cauchy Distribution

The Cauchy distribution is a continuous probability distribution and it has two parameters x_0 and γ. x_0 is the peak location of the distribution and γ stands for the halfwidth at halfmaximum (HWHM) of the distribution. The value of γ determines the shape of the Cauchy distribution. If γ is assigned a lower value, the peak of the probability density function would be higher and its width would be narrower. On the other hand, if γ is assigned a higher value, the probability density function would have a lower peak and a wider width. The Cauchy distribution generates a large step from the peak with a higher probability. In general, many evolutionary algorithms have used this long tail property as an escaping local minima technique. The probability density function and the cumulative distribution function of the Cauchy distribution are defined by

$$f(x; x_0, \gamma) = \frac{1}{\pi\gamma\left[1 + \left((x - x_0)/\gamma\right)^2\right]} = \frac{1}{\pi}\left[\frac{\gamma}{(x - x_0)^2 + \gamma^2}\right], \quad (12)$$

$$F(x; x_0, \gamma) = \frac{1}{\pi}\arctan\left((x - x_0)/\gamma\right) + \frac{1}{2}.$$

Figure 1 illustrates the various probability density functions of the Cauchy distribution. Here, L and S denote the location (x_0) and the scaling factor (γ), respectively. In addition, $L = 0$ and $S = 1$ generate the standard Cauchy distribution.

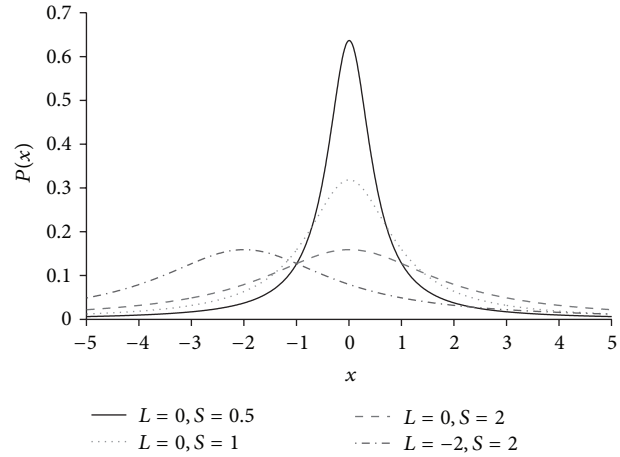


Figure 1: The various probability density functions of the Cauchy distribution.

4. Adaptive Cauchy DE

4.1. When Parameter Adaptation Should Be Performed? Finding appropriate moments of adapting control parameters is important problem for improving the DE performance. In this section, we explain when parameter adaptation should be performed.

Looking for previous studies, jDE utilizes self-adaptive method which allows each individual to maintain suitable control parameter values by itself. However, the parameter adaptation of jDE depends on the predefined probabilities (τ_1 and τ_2). Therefore, this method does not guarantee the adequacy of maintained control parameter values for current generation. In other words, it is possible that some individuals maintain unsuitable control parameter values. In SaDE, the scaling factor is calculated in every generation by using Gaussian distribution with the predefined mean value and all individuals utilize it. The crossover rate of SaDE, each individual has its own crossover rate and they are calculated by using Gaussian distribution with the median value of accumulated information about selection operation results during learning period as a mean value. This method is performed in every end of learning period. The parameter adaptation of SaDE has two problems. First, although each individual has different state during the DE iteration, the scaling factor adaptation of SaDE does not consider it. Therefore, many individuals might be utilized unsuitable control parameter values. Second, the selection operation results of past generations might become unnecessary or even noisy information for adapting the crossover rate. In addition, similar to jDE, during the learning period, it is possible that some individuals maintain unsuitable control parameter values.

Parameter adaptation should be performed whenever current control parameter values are not suitable for finding optimal value. We can utilize the selection operation results for distinguishing that an individual has suitable control parameters or not because the DE is based on the elitism. We find out that parameter adaptation should be performed in every generation. This means that every generation is appropriate moments of adapting control parameters. The reasons are follows.

(1) If an individual has good control parameter values, the child individual can succeed in the selection operation and it can locate better region for finding optimal value than the region of its parent individual. At this moment, the characteristic of child individual region might differ from the characteristic of its parent region. It means that there is a possibility of the existing of more suitable control parameter values than the previous control parameter values for new region. Therefore, although an individual succeeds in the selection operation, we should apply parameter adaptation for finding more suitable control parameter values.

(2) On the contrary, if an individual does not have good parameter values, the child individual might fail to evolve in the selection operation and then it remains the same region with its parent individual. This indicates that the individual needs more suitable control parameter values for escaping the region. Therefore, if an individual fails to evolve itself, we should also apply parameter adaptation.

Table 1: Benchmark functions used in the performance evaluation.

Benchmark function	Dim	Search space	Global min.
$F_1(x) = \sum_{i=1}^{D} x_i^2$	30	$[-100, 100]^D$	0
$F_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	30	$[-10, 10]^D$	0
$F_3(x) = \max_i(\lvert x_i \rvert, 1 \le i \le D)$	30	$[-100, 100]^D$	0
$F_4(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	30	$[-100, 100]^D$	0
$F_5(x) = \sum_{i=1}^{D} i x_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]^D$	0
$F_6(x) = \sum_{i=1}^{D} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	$[-500, 500]^D$	−12569.5
$F_7(x) = \sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$	30	$[-5.12, 5.12]^D$	0
$F_8(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i\right) + 20 + \exp(1)$	30	$[-32, 32]^D$	0
$F_9(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^D$	0
$F_{10}(x) = \frac{\pi}{D}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_D - 1)^2\right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$	30	$[-50, 50]^D$	0
$F_{11}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2\left[1 + \sin^2(3\pi x_{i+1})\right] + (x_D - 1)^2\left[1 + \sin^2(2\pi x_D)\right]\right\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	30	$[-50, 50]^D$	0
$F_{12}(x) = f_{12}(x_n, x_1) + \sum_{i=1}^{D-1} f_{12}(x_i, x_{i+1})$ $f_{12}(x, y) = (x^2 + y^2)^{0.25}\left[\sin^2\left(50(x^2 + y^2)^{0.1}\right) + 1\right]$	30	$[-100, 100]^D$	0
$F_{13}(x) = \sum_{i=1}^{D-1}\left(x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7\right)$	30	$[-15, 15]^D$	0
$F_{14}(x) = \sum_{i=1}^{D-1}\left(x_i^2 + x_{i+1}^2\right)^{0.25}\left[\sin^2\left(50\left(x_i^2 + x_{i+1}^2\right)^{0.1}\right) + 1\right]$	30	$[-100, 100]^D$	0

As a result, because the individuals of DE are evolved for exploring new regions, it is hard to assure that the previous suitable control parameter values are still suitable until satisfying some probabilities or during some periods. Therefore, parameter adaptation should be performed in every generation.

4.2. How Parameter Adaptation Should Be Performed? Finding proper method of adapting control parameters is also important problem for improving the DE performance. In this section, we explain how parameter adaptation should be performed. When performing parameter adaptation, we can utilize the successfully evolved individuals' control parameter values for parameter adaptation. It is because the successfully evolved individuals are led by good parameter values. That is to say, good parameter values make the individuals take the good region for solving problem.

Looking for previous studies, jDE adapts control parameter values by using the uniform distribution. However, the randomly generated control parameter values might not be suitable for finding better region. In SaDE, the successfully evolved individuals' crossover rates are stored in CR_Memory. After learning period, the parameter adaptation of SaDE extracts the median value from the CR_Memory as a mean value of the Gaussian distribution. In general, the median function is not largely influenced by outliers. However, the outliers give us the information about a new possibility of better control parameter values. Therefore, we should consider the outliers together.

As a result, the successfully evolved individuals' control parameter values based parameter adaptation is proper

Table 2: The experiment result of comparison of adaptive Cauchy DE with other DE algorithms.

	GEN	Adaptive Cauchy DE		DE/rand/1/bin		jDE		SaDE		MDE	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	1500	**5.0E − 36**	**9.4E − 36**	7.9E − 14	6.8E − 14	2.6E − 28	4.0E − 28	1.8E − 20	2.3E − 20	7.0E − 17	2.8E − 17
F_2	2000	**2.4E − 30**	**1.6E − 30**	1.2E − 09	6.7E − 10	1.8E − 23	1.8E − 23	6.2E − 15	3.2E − 15	4.8E − 13	1.3E − 13
F_3	5000	2.9E − 12	1.1E − 12	3.3E − 02	7.2E − 02	**2.0E − 15**	**3.3E − 15**	7.8E − 10	2.0E − 10	2.0E − 08	8.5E − 09
F_4	1500	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00
F_5	3000	**3.0E − 03**	**6.9E − 04**	4.6E − 03	1.0E − 03	3.1E − 03	8.5E − 04	4.5E − 03	1.2E − 03	8.8E − 03	1.8E − 03
F_6	9000	**−12569.5**	**7.3E − 12**	−11095.3	5.2E + 02	**−12569.5**	**7.3E − 12**	**−12569.5**	**7.3E − 12**	−11482.1	3.0E + 02
F_7	5000	**0.0E + 00**	**0.0E + 00**	7.1E + 01	2.9E + 01	**0.0E + 00**	**0.0E + 00**	**0.0E + 00**	**0.0E + 00**	4.0E + 01	5.6E + 00
F_8	1500	**3.3E − 15**	**7.0E − 16**	9.2E − 08	4.0E − 08	8.2E − 15	2.3E − 15	5.3E − 11	3.7E − 11	4.0E − 09	9.2E − 10
F_9	2000	**0.0E + 00**	**0.0E + 00**	3.9E − 04	2.0E − 03	**0.0E + 00**	**0.0E + 00**	**0.0E + 00**	**0.0E + 00**	7.4E − 03	1.1E − 02
F_{10}	1500	**1.6E − 32**	**5.5E − 48**	6.5E − 15	5.6E − 15	7.0E − 30	7.2E − 30	3.3E − 20	4.2E − 20	9.9E − 18	1.5E − 17
F_{11}	1500	**1.3E − 32**	**1.1E − 47**	5.9E − 14	4.9E − 14	1.2E − 28	1.4E − 28	8.5E − 20	1.6E − 19	2.6E − 17	1.6E − 17
F_{12}	3000	**6.1E − 22**	**5.0E − 22**	2.1E − 07	1.3E − 07	6.3E − 17	7.0E − 17	3.5E − 12	2.9E − 12	7.4E − 11	3.9E − 11
F_{13}	1000	**5.0E − 20**	**3.4E − 20**	7.3E − 04	6.6E − 04	1.4E − 15	6.9E − 16	1.5E − 07	3.4E − 08	3.4E − 05	1.3E − 05
F_{14}	3000	**3.5E − 22**	**2.5E − 22**	2.3E − 05	1.7E − 05	7.5E − 17	1.4E − 16	1.8E − 10	2.2E − 10	1.2E − 08	3.7E − 09

Table 3: The success rate of comparison of adaptive Cauchy DE with other DE algorithms.

Success rate	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}
Adaptive Cauchy DE	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
DE/rand/1/bin	100%	100%	38%	100%	100%	0%	0%	100%	96%	100%	100%	100%	0%	18%
jDE	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
SaDE	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
MDE	100%	100%	100%	100%	72%	0%	0%	100%	58%	100%	100%	100%	0%	100%

Table 4: The experiment result of comparison of adaptive Cauchy DE with FEP and CEP.

	GEN	Adaptive Cauchy DE		DE/rand/1/bin		FEP		CEP	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	1500	**5.0E − 36**	**9.4E − 36**	7.9E − 14	6.8E − 14	5.7E − 04	1.3E − 04	2.2E − 04	5.9E − 04
F_2	2000	**2.4E − 30**	**1.6E − 30**	1.2E − 09	6.7E − 10	8.1E − 03	7.7E − 04	2.6E − 03	1.7E − 04
F_3	5000	**2.9E − 12**	**1.1E − 12**	3.3E − 02	7.2E − 02	3.0E − 01	5.0E − 01	2.0E + 00	1.2E + 00
F_4	1500	**0.0E + 00**	**0.0E + 00**	**0.0E + 00**	**0.0E + 00**	**0.0E + 00**	**0.0E + 00**	5.8E + 02	1.1E + 03
F_5	3000	**3.0E − 03**	**6.9E − 04**	4.6E − 03	1.0E − 03	7.6E − 03	2.6E − 03	1.8E − 02	6.4E − 03
F_6	9000	**−12569.5**	**7.3E − 12**	−11095.3	5.2E + 02	−12554.5	5.3E + 01	−7917.1	6.3E + 02
F_7	5000	**0.0E + 00**	**0.0E + 00**	7.1E + 01	2.9E + 01	4.6E − 02	1.2E − 02	8.9E + 01	2.3E + 01
F_8	1500	**3.3E − 15**	**7.0E − 16**	9.2E − 08	4.0E − 08	1.8E − 02	2.1E − 03	9.2E + 00	2.8E + 00
F_9	2000	**0.0E + 00**	**0.0E + 00**	3.9E − 04	2.0E − 03	1.6E − 02	2.2E − 02	8.6E − 02	1.2E − 01
F_{10}	1500	**1.6E − 32**	**5.5E − 48**	6.5E − 15	5.6E − 15	9.2E − 06	3.6E − 06	1.8E + 00	2.4E + 00
F_{11}	1500	**1.3E − 32**	**1.1E − 47**	5.9E − 14	4.9E − 14	1.6E − 04	7.3E − 05	1.4E + 00	3.7E + 00

method to adapting control parameter values. This method should also consider the outliers.

4.3. The Proposed Algorithm. The proposed algorithm makes use of DE/rand/1/bin as a basic framework, in which the mutation is one of weaker greedy mutation strategies. In general, this mutation strategy is not so efficient in solving the unimodal problems since its lack of the fast convergence property makes the population slowly converge into the global minimum. However, if the control parameters are adapted suitably, this strategy can also demonstrate a good performance property in the unimodal and the multimodal problems.

The proposed algorithm adjusts two control parameters, F and CR, except for NP. The control parameter NP does not seriously affect the performance of DE more than the other two control parameters. Prior to explaining the adaptation procedures, the characteristics of these parameters are described. The control parameter F is related to the convergence speed of DE. Therefore, a higher value of F encourages the exploration power which is generally useful in the early stage of DE. On the other hand, a lower value of

Table 5: The experiment result of comparison of adaptive Cauchy DE with adaptive LEP and best Lévy.

	GEN	Adaptive Cauchy DE		DE/rand/1/bin		Adaptive LEP		Best Lévy	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	1500	**4.7E − 36**	**5.1E − 36**	7.2E − 14	7.9E − 14	6.3E − 04	7.6E − 05	6.6E − 04	6.4E − 05
F_6	1500	**−12569.5**	**7.3E − 12**	−6506.71	6.7E + 02	−11469.2	5.8E + 01	−11898.2	5.2E + 01
F_7	1500	**0.0E + 00**	**0.0E + 00**	1.7E + 02	1.2E + 01	5.9E + 00	2.1E + 00	1.3E + 01	2.3E + 00
F_8	1500	**3.2E − 15**	**5.0E − 16**	9.1E − 08	3.7E − 08	1.9E − 02	1.0E − 03	3.1E − 02	2.0E − 03
F_9	1500	**0.0E + 00**	**0.0E + 00**	2.2E − 13	1.4E − 13	2.4E − 02	2.8E − 02	1.8E − 02	1.7E − 02
F_{10}	1500	**1.6E − 32**	**5.5E − 48**	7.5E − 15	7.2E − 15	6.0E − 06	1.0E − 06	3.0E − 05	4.0E − 06
F_{11}	1500	**1.3E − 32**	**1.1E − 47**	5.4E − 14	4.9E − 14	9.8E − 05	1.2E − 05	2.6E − 04	3.0E − 05

Table 6: The experiment result of comparison of various failure counters.

	GEN	$FC_F = 0$, $FC_{CR} = 0$		$FC_F = 0$, $FC_{CR} = 1$		$FC_F = 1$, $FC_{CR} = 0$		$FC_F = 1$, $FC_{CR} = 1$		$FC_F = 2$, $FC_{CR} = 2$	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	1500	5.0E − 36	9.4E − 36	4.0E − 31	5.2E − 31	**6.9E − 40**	**6.8E − 40**	2.1E − 35	4.1E − 35	1.4E − 30	2.0E − 30
F_2	2000	2.4E − 30	1.6E − 30	1.7E − 26	1.5E − 26	**8.7E − 34**	**5.2E − 34**	3.4E − 30	3.1E − 30	2.0E − 26	1.4E − 26
F_3	5000	**2.9E − 12**	**1.1E − 12**	1.9E − 04	3.0E − 05	2.5E + 00	2.9E + 00	4.7E − 01	9.7E − 01	1.1E + 00	1.5E + 00
F_4	1500	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00
F_5	3000	**3.0E − 03**	**6.9E − 04**	3.1E − 03	8.4E − 04	3.1E − 03	8.9E − 04	3.0E − 03	8.0E − 04	3.1E − 03	8.3E − 04
F_6	9000	**−12569.5**	**7.3E − 12**	**−12569.5**	**7.3E − 12**	**−12569.5**	**7.3E − 12**	**−12569.5**	**7.3E − 12**	−12567.1	1.7E + 01
F_7	5000	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00
F_8	1500	**3.3E − 15**	**7.0E − 16**	1.4E − 14	4.5E − 15	3.1E − 15	0.0E + 00	3.0E − 07	2.1E − 06	1.5E − 14	2.6E − 15
F_9	2000	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00
F_{10}	1500	**1.6E − 32**	**5.5E − 48**	9.6E − 32	6.6E − 32	4.8E − 07	3.4E − 06	1.6E − 32	3.6E − 34	1.7E − 31	1.6E − 31
F_{11}	1500	**1.3E − 32**	**1.1E − 47**	7.3E − 31	1.1E − 30	1.4E − 32	6.9E − 34	1.3E − 32	1.1E − 47	8.5E − 31	7.8E − 31
F_{12}	3000	6.1E − 22	5.0E − 22	1.6E − 18	1.8E − 18	**1.4E − 23**	**6.9E − 23**	2.0E − 21	2.1E − 21	1.4E − 18	1.4E − 18
F_{13}	1000	5.0E − 20	3.4E − 20	1.7E − 16	1.4E − 16	**5.8E − 23**	**3.3E − 23**	8.6E − 20	5.9E − 20	4.7E − 17	4.1E − 17
F_{14}	3000	3.5E − 22	2.5E − 22	7.2E − 19	7.1E − 19	**9.5E − 25**	**1.1E − 24**	8.0E − 22	8.9E − 22	8.3E − 19	6.6E − 19

F promotes the exploitation power that is usually desirable in the later stage of DE. Moreover, the value of control parameter CR is related to the diversity of population.

The parameter adaptation of proposed algorithm utilizes F_Memory and CR_Memory. The successfully evolved individuals' scaling factors and crossover rates are stored in these memories. When performing parameter adaptation, arithmetic mean function is applied to extract mean values and these are actual parameter values of the Cauchy distribution as location parameters. The Cauchy distribution is one of the long tail distributions. The Cauchy distribution generates the large step from the peak location with higher probability. There is a possibility that the current appropriate parameter values might be the inappropriate parameter values in next generation. Therefore, we cannot assure that the average parameter value is still the well-suited parameter value for the future generations. In view of the above considerations, the parameter adaptation of the proposed algorithm utilizes the Cauchy distribution as a large step method. Through this, the control parameters of each individual are assigned either near the average parameter value or far from that of the average parameter value which might be the better parameter value of the next generation.

The details of the proposed algorithm are given as follows. First of all, all individuals have their own control parameters, F_i and CR_i where i is the individual's index. At the initialization stage, these parameters are initialized as 0.5 and 0.9, respectively. The mutation and crossover operations used in DE/rand/1/bin are employed. In the selection operation, if the trial vector is selected as an individual for the next generation, the control parameter values of this individual are stored in the F_Memory and CR_Memory. After the selection operation, the parameter adaptation is carried out.

The parameter F_i is adapted by the Cauchy distribution with the average parameter value. After that, the F_i is truncated to 0.1 or 1 if the F_i is less than 0.1 or greater than 1. The adaptation of the scaling factor is performed as follows:

$$F_{i,G+1} = C\left(0, \gamma_F\right) + F_{\text{avg},G}, \tag{13}$$

where $F_{\text{avg},G}$ is the average parameter value of the accumulated information in the F_Memory as the location parameter of the Cauchy distribution. The γ_F is scaling factor of the equation and is assigned 0.1.

Similarly, the CR_i is adapted by the Cauchy distribution with the average parameter value. After that, the CR_i is

truncated to 0 or 1 if the CR_i is less than 0 or greater than 1. The adaptation of the crossover rate is given as follows:

$$CR_{i,G+1} = C\left(0, \gamma_{CR}\right) + CR_{avg,G}, \quad (14)$$

where $CR_{avg,G}$ is the average parameter value of the accumulated information in the CR_Memory as the location parameter of the Cauchy distribution. The γ_F is scaling factor of the equation and is assigned 0.1. Algorithm 1 describes the pseudocode of the proposed algorithm.

When performing parameter adaptation, if there is no successfully evolved individual, then the average parameter values are assigned the average parameter values of last generation.

5. Performance Evaluation

5.1. Benchmark Functions. The performance of proposed algorithm was evaluated by fourteen benchmark functions. The first eleven benchmark functions are from [14, 15] and the rest benchmark functions are Extended f_{12} (F_{12}), Bohachevsky (F_{13}), and Schaffer (F_{14}). The functions are shown in Table 1.

The characteristics of the benchmark functions are described as follows: F_1–F_3 are continuous unimodal functions, F_4 is a discontinuous step function, F_5 is a noise quadratic function, and F_6–F_{14} are continuous multimodal functions that the number of local minima exponentially increases when their dimension grows. A more detailed description of each function is given in [14, 15].

5.2. Experiment Setup. The proposed algorithm was compared to standard DE algorithm and several state-of-art adaptive DE algorithms. The five algorithms in comparison are listed as follows:

(1) adaptive Cauchy DE;

(2) DE/rand/1/bin with $F = 0.5$ and $CR = 0.9$ [1, 4, 16, 17];

(3) jDE [4];

(4) SaDE [7];

(5) MDE with MFC = 5 [9].

All of the used parameter values are the recommended or utilized parameter values by their authors. The population size NP is fixed by 100 in all experiments. The maximum number of generations is assigned by 1500 for F_1, F_4, F_8, F_{10}, and F_{11}; 2000 for F_2 and F_9; 3000 for F_5, F_{12}, and F_{14}; 5000 for F_3 and F_7; 9000 for F_6; 1000 for F_{13}. All experiment results were run 50 times, independently. For clarity, the result of the best algorithm is marked in boldface. If the difference between the global minimum and the best fitness is lower than 10^{-5} (In F_5, 10^{-2}), we countered the experiment is successful.

5.3. Comparison of Adaptive Cauchy DE with Adaptive DE Algorithms. The mean and the standard deviation of experiment results obtained by adaptive Cauchy DE and the compared DE algorithms for F_1–F_{14} for $D = 30$ are summarized in Table 2.

The proposed algorithm shows better performance on solving the unimodal problems as well as in the multimodal problems except F_3 benchmark function. jDE shows the best performance in F_3 benchmark function. The proposed

```
/* Initialization */
Generate the initial population
Evaluate the initial population
FOR i = 0 to NP DO
   F_i = 0.5
   CR_i = 0.9
END FOR

WHILE The termination condition is not satisfied DO
   /* Mutation operation */
   FOR i = 0 to NP DO
      Generate a mutant vector V_{i,G}
      Randomly select three donor vectors X_{r1}, X_{r2}, X_{r3}
      V_{i,G} = X_{r1,G} + F_i · (X_{r2,G} − X_{r3,G})
   END FOR

   /* Crossover operation */
   FOR i = 0 to NP DO
      Generate a trial vector U_{i,G}
      Select a random number j_rand lying between [1, D]
      FOR j = 0 to D DO
         IF rand[0, 1) ≤ CR_i or j == j_rand THEN
            u_{i,j,G} = v_{i,j,G}
         ELSE
            u_{i,j,G} = x_{i,j,G}
         END IF
      END FOR
   END FOR

   /* Selection operation */
   k = 0
   FOR i = 0 to NP DO
      IF f(U_{i,G}) ≤ f(X_{i,G}) THEN
         X_{i,G+1} = U_{i,G}
         F_Memory[k] = F_i
         CR_Memory[k] = CR_i
         k+ = 1
      ELSE
         X_{i,G+1} = X_{i,G}
      END IF
   END FOR

   /* Parameter adaptation */
   IF k ≠ 0 THEN
      F_{avg,G} = mean(F_Memory, k)
      CR_{avg,G} = mean(CR_Memory, k)
   END IF
   FOR i = 0 to NP DO
      F_{i,G+1} = C(0, γ_F) + F_{avg,G}
      CR_{i,G+1} = C(0, γ_CR) + CR_{avg,G}
   END FOR
END WHILE
```

Algorithm 1: Adaptive Cauchy DE.

Table 7: The success rate of comparison of various failure counters.

Success rate	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}
$FC_F = 0$, $FC_{CR} = 0$	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
$FC_F = 0$, $FC_{CR} = 1$	100%	100%	0%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
$FC_F = 1$, $FC_{CR} = 0$	100%	100%	0%	100%	100%	100%	100%	100%	100%	98%	100%	100%	100%	100%
$FC_F = 1$, $FC_{CR} = 1$	100%	100%	0%	100%	100%	100%	100%	98%	100%	100%	100%	100%	100%	100%
$FC_F = 2$, $FC_{CR} = 2$	100%	100%	0%	100%	100%	98%	100%	100%	100%	100%	100%	100%	100%	100%

Table 8: The experiment result of comparison of various mathematical functions for utilizing success memories.

	GEN	Arithmetic mean		Median		Best individual		Itself	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	1500	**5.0E − 36**	**9.4E − 36**	2.9E − 27	4.0E − 27	2.4E − 31	8.2E − 31	4.2E − 19	2.5E − 19
F_2	2000	**2.4E − 30**	**1.6E − 30**	5.5E − 23	4.4E − 23	8.1E − 28	2.4E − 27	2.9E − 16	1.2E − 16
F_3	5000	**2.9E − 12**	**1.1E − 12**	2.1E + 00	2.5E + 00	8.1E + 00	7.5E + 00	4.7E − 04	3.3E − 03
F_4	1500	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00
F_5	3000	**3.0E − 03**	**6.9E − 04**	4.6E − 03	1.2E − 03	4.1E − 03	1.4E − 03	4.4E − 03	1.1E − 03
F_6	9000	**−12569.5**	**7.3E − 12**	**−12569.5**	**7.3E − 12**	**−12569.5**	**7.3E − 12**	−12567.1	1.7E + 01
F_7	5000	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00	0.0E + 00
F_8	1500	**3.3E − 15**	**7.0E − 16**	3.0E − 14	8.4E − 15	7.4E − 15	2.7E − 14	1.7E − 10	6.7E − 11
F_9	2000	**0.0E + 00**	**0.0E + 00**	0.0E + 00	0.0E + 00	3.5E − 04	1.7E − 03	3.9E − 04	2.0E − 03
F_{10}	1500	**1.6E − 32**	**5.5E − 48**	4.8E − 29	4.1E − 29	5.0E + 00	3.5E + 01	1.2E − 20	1.2E − 20
F_{11}	1500	**1.3E − 32**	**1.1E − 47**	8.0E − 28	1.1E − 27	1.1E − 02	7.5E − 02	1.4E − 19	1.2E − 19
F_{12}	3000	**6.1E − 22**	**5.0E − 22**	1.1E − 16	9.2E − 17	4.1E + 03	2.7E + 04	4.0E − 11	2.8E − 11
F_{13}	1000	**5.0E − 20**	**3.4E − 20**	2.3E − 15	8.7E − 16	1.3E − 02	3.5E − 02	1.2E − 10	5.1E − 11
F_{14}	3000	**3.5E − 22**	**2.5E − 22**	1.5E − 15	1.8E − 15	2.3E − 01	5.3E − 01	1.1E − 09	6.6E − 10

algorithm outperformed all multimodal problems. The second best algorithm is jDE. Although SaDE utilizes strategy adaptation as well as parameter adaptation, jDE shows better results than SaDE in all benchmark functions. It means that parameter adaptation is more important to improve the performance of DE. MDE shows better performance than DE/rand/1/bin in several unimodal and multimodal problems.

Table 3 shows the success rate of comparison results. The success rate is obtained by a mount of successful counter divided by a mount of experiment runs (50). The proposed algorithm and two adaptive DE algorithms (jDE and SaDE) show perfect success rates. However, DE/rand/1/bin and MDE show lower success rate than the proposed algorithm and they totally failed to find global optimum in several benchmark functions.

Figure 2 shows the average best graphs of adaptive Cauchy DE and the compared DE algorithms.

5.4. Comparison of Adaptive Cauchy DE and FEP and CEP. The mean deviation and the standard deviation of experiment results obtained by adaptive Cauchy DE, DE/rand/1/bin, FEP (Fast Evolutionary Programming), and CEP (Classic Evolutionary Programming) for F_1–F_{11} for $D = 30$ are summarized in Table 4. The results of FEP and CEP are taken from [13, Tables 2-4].

The proposed algorithm shows better performance on solving all benchmark functions than DE/rand/1/bin, FEP, and CEP. The second best algorithm is DE/rand/1/bin. However, DE/rand/1/bin shows lower performance than FEP in several benchmark functions (F_6 and F_7).

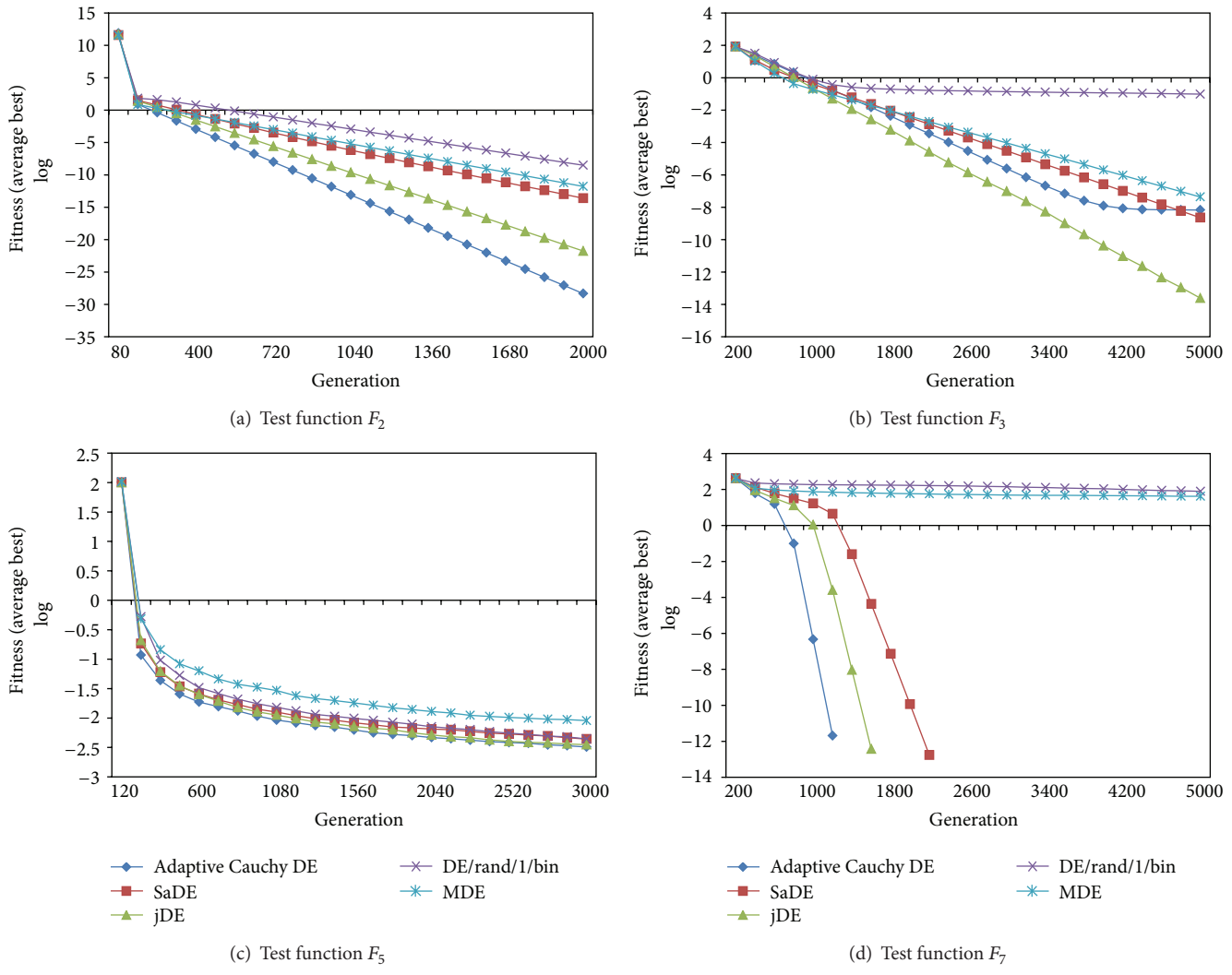
5.5. Comparison of Adaptive Cauchy DE and Adaptive LEP and Best Lévy. The mean deviation and the standard deviation of experiment results obtained by adaptive Cauchy DE, DE/rand/1/bin, adaptive LEP, and best Lévy for F_1 and F_6–F_{11} for $D = 30$ are summarized in Table 5. The results of adaptive LEP and best Lévy are taken from [18, Table 3]. The population size NP is fixed by 100 in all experiments. The maximum number of generations is assigned by 1500 for all benchmark functions.

The proposed algorithm shows better performance on solving all benchmark functions than DE/rand/1/bin, FEP, and CEP. The second best algorithm is DE/rand/1/bin again. However, DE/rand/1/bin shows lower performance than adaptive LEP and best Lévy in several benchmark functions (F_6 and F_7).

5.6. Parameter Study. Tables 6 and 7 show that various failure counter experiment results. The goal of this experiments is finding appropriate moments of parameter adaptation. $FC_F = n$ means if an individual fails to evolve itself consequently n

Table 9: The success rate of comparison of various mathematical functions for utilizing success memories.

Success rate	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}
Arithmetic mean	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Median	100%	100%	0%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Best individual	100%	100%	0%	100%	100%	100%	100%	100%	96%	90%	96%	92%	74%	82%
Itself	100%	100%	90%	100%	100%	98%	100%	100%	96%	100%	100%	100%	100%	100%



(a) Test function F_2

(b) Test function F_3

(c) Test function F_5

(d) Test function F_7

Figure 2: Average best graphs of Adaptive Cauchy DE with the compared DE algorithms.

times, the scaling factor of individual is adapted. Similarly, $FC_{CR} = n$ means if an individual fails to evolve itself consequently n times, the crossover rate of individual is adapted. For example, if $FC_F = 1$ and $FC_{CR} = 0$, then an individual's scaling factor is adapted when the individual fails to evolve itself, the last selection operation and the crossover of individual aer adapted in every generation.

The results show that adapting control parameters $FC_F = 0$ with $FC_{CR} = 0$ and $FC_F = 1$ with $FC_{CR} = 0$ had good performance in the comparison. However, when comparing success rate, $FC_F = 0$ with $FC_{CR} = 0$ had higher success rate than $FC_F = 1$ with $FC_{CR} = 0$ in F_3 benchmark function.

Note that when failure counter is increasing, the performance of algorithm decreased. Therefore, parameter adaptation should be performed in every generation. This is because the individuals of DE are evolved for exploring new regions. Therefore, it is hard to assure that the previous suitable control parameter values are still suitable until satisfying some probabilities or during some periods.

Tables 8 and 9 show the various parameter adaptation method experiment results. The goal of these experiments is finding proper method of utilizing F_Memory and CR_Memory for parameter adaptation. Arithmetic mean indicates that the proposed algorithm utilized arithmetic

Table 10: The experiment result of comparison of Cauchy distribution with Gaussian distribution for parameter adaptation.

	GEN	Cauchy $\gamma = 0.1$		Cauchy $\gamma = 0.3$		Gaussian Std = 0.1		Gaussian Std = 0.3	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	1500	**5.0E − 36**	**9.4E − 36**	$1.6E-28$	$1.2E-28$	$2.8E-32$	$3.4E-32$	$2.8E-30$	$2.8E-30$
F_2	2000	**2.4E − 30**	**1.6E − 30**	$3.1E-24$	$1.3E-24$	$1.9E-27$	$2.1E-27$	$7.4E-26$	$4.7E-26$
F_3	5000	**2.9E − 12**	**1.1E − 12**	$1.3E-02$	$6.7E-02$	$1.3E-01$	$5.6E-01$	$5.3E-12$	$9.0E-12$
F_4	1500	$0.0E+00$	$0.0E+00$	$0.0E+00$	$0.0E+00$	$0.0E+00$	$0.0E+00$	$0.0E+00$	$0.0E+00$
F_5	3000	**3.0E − 03**	**6.9E − 04**	$4.0E-03$	$1.3E-03$	$3.4E-03$	$1.6E-03$	$3.1E-03$	$8.4E-04$
F_6	9000	−12569.5	$7.3E-12$	−12569.5	$7.3E-12$	−12569.5	$7.3E-12$	−12569.5	$7.3E-12$
F_7	5000	$0.0E+00$	$0.0E+00$	$0.0E+00$	$0.0E+00$	$0.0E+00$	$0.0E+00$	$0.0E+00$	$0.0E+00$
F_8	1500	**3.3E − 15**	**7.0E − 16**	$8.7E-15$	$2.7E-15$	$4.2E-15$	$1.6E-15$	$6.3E-15$	$1.1E-15$
F_9	2000	**0.0E + 00**	**0.0E + 00**	$2.0E-04$	$1.4E-03$	**0.0E + 00**	**0.0E + 00**	**0.0E + 00**	**0.0E + 00**
F_{10}	1500	**1.6E − 32**	**5.5E − 48**	$6.2E-30$	$4.9E-30$	$1.9E-32$	$6.7E-33$	$2.9E-31$	$1.9E-31$
F_{11}	1500	**1.3E − 32**	**1.1E − 47**	$1.0E-28$	$9.4E-29$	$6.0E-32$	$7.6E-32$	$2.3E-30$	$2.8E-30$
F_{12}	3000	**6.1E − 22**	**5.0E − 22**	$3.7E-17$	$2.9E-17$	$1.5E-19$	$2.1E-19$	$3.5E-18$	$2.6E-18$
F_{13}	1000	**5.0E − 20**	**3.4E − 20**	$2.7E-16$	$1.1E-16$	$1.6E-17$	$8.0E-18$	$6.7E-17$	$2.8E-17$
F_{14}	3000	**3.5E − 22**	**2.5E − 22**	$4.4E-17$	$6.6E-17$	$2.1E-19$	$2.4E-19$	$2.9E-18$	$2.5E-18$

Table 11: The success rate of comparison Cauchy distribution with Gaussian distribution for parameter adaptation.

Success rate	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}
Cauchy $\gamma = 0.1$	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Cauchy $\gamma = 0.3$	100%	100%	78%	100%	100%	100%	100%	100%	98%	100%	100%	100%	100%	100%
Gaussian Std = 0.1	100%	100%	72%	100%	98%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Gaussian Std = 0.3	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

mean function to extract mean values from F_Memory and CR_Memory and each individual's control parameter values are adapted based on the mean values as location parameters of the Cauchy distribution for parameter adaptation. Similarly, median indicates that the proposed algorithm utilized median function to extract median values from F_Memory and CR_Memory and each individual's control parameter values are adapted based on the median values as location parameters of the Cauchy distribution for parameter adaptation. Best individual indicates that each individual's control parameter values are adapted based on the best individual's control parameter values as location parameter of the Cauchy distribution. Finally, itself indicates that each individual's control parameter values are adapted based on its own control parameter values as location parameter of the Cauchy distribution.

The results show that adapting control parameters based on arithmetic mean function had good performance than median function. This is because the outliers give us the information about a new possibility of better control parameter values. Therefore, the arithmetic mean function is more applicable than median function. Parameter adaptation based on its own control parameter values shows good success rate in the comparison. However, the performance was lower than that of other methods. Parameter adaptation based on best individual's control parameter values shows good performance in only unimodal problems.

Tables 10 and 11 show the comparison results of the Cauchy distribution with the Gaussian distribution for parameter adapation method. The goal of these experiments is finding which distribution property (short or long tail) is more suitable for parameter adaptation. Cauchy $\gamma = 0.1$ indicates that parameter adaptation is performed based on the Cauchy distribution and the scaling parameter of distribution is assigned 0.1. Gaussian Std = 0.1 indicates that parameter adaptation is performed based on the Gaussian distribution and the standard deviation parameter of distribution is assigned 0.1.

The experiment results show that the Cauchy distribution with $\gamma = 0.1$ had good performance than others. This is because, the Cauchy distribution generates the large step from the peak location with higher probability. Therefore, the control parameters of each individual are assigned either near the average parameter value or far from that of the average parameter value which might be the better parameter value of the next generation.

6. Conclusion

The parameters of DE should be adequately assigned to attain better performance. But finding suitable values demands a lot of computational resources. In this sense, we present a new DE algorithm which utilizes success memories of scaling factors and crossover rates to properly adjust the control parameters of DE; the control parameters are adapted at each generation based on the Cauchy distribution with mean values of success memories. Experimental results showed that the adaptive Cauchy DE algorithm generally achieves

better performance than existing DE variants on various multimodal and unimodal test problems. The results also supported the claim that a long tail distribution is more reliable than a short tail distribution in adjusting the control parameters.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea government (MSIP) (no. 2012-013-735). This work was also supported by the DGIST R&D Program of the Ministry of Education, Science and Technology of Korea (13-BD-01).

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] R. Gamperle, S. D. Muller, and P. Koumoutsakos, "A parameter study for differential evolution," in *Proceedings of the International Conference on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation (WSEAS '02)*, pp. 293–298, 2002.

[3] J. Zhang and A. C. Sanderson, "An approximate Gaussian model of differential evolution with spherical fitness functions," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 2220–2228, September 2007.

[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 Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[5] Á. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.

[6] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Springer, Berlin, Germany, 2003.

[7] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '05)*, vol. 2, pp. 1785–1791, September 2005.

[8] J. Brest, B. Bošković, S. Greiner, V. Žumer, and M. S. Maučec, "Performance comparison of self-adaptive and adaptive differential evolution algorithms," *Soft Computing*, vol. 11, no. 7, pp. 617–629, 2007.

[9] M. Ali and M. Pant, "Improving the performance of differential evolution algorithm using Cauchy mutation," *Soft Computing*, vol. 15, no. 5, pp. 991–1007, 2011.

[10] J. Teo, "Exploring dynamic self-adaptive populations in differential evolution," *Soft Computing*, vol. 10, no. 8, pp. 673–686, 2006.

[11] H. A. Abbass, "The self-adaptive Pareto differential evolution algorithm," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, pp. 831–836, May 2002.

[12] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[13] J. Zhang and A. C. Sanderson, *Adaptive Differential Evolution: A Robust Approach to Multimodal Problem Optimization*, Springer, Berlin, Germany, 2009.

[14] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.

[15] X. Yao, Y. Liu, K. H. Liang, and G. Lin, "Fast evolutionary algorithms," in *Advances in Evolutionary Computing*, pp. 45–94, Springer, New York, NY, USA, 2003.

[16] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," in *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference (GECCO '06)*, pp. 485–492, July 2006.

[17] J. Vesterstrøm and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, vol. 2, pp. 1980–1987, June 2004.

[18] E. Cuevas, D. Zaldivar, and M. Pérez-Cisneros, "A novel multi-threshold segmentation approach based on differential evolution optimization," *Expert Systems with Applications*, vol. 37, no. 7, pp. 5265–5271, 2010.