An Adaptive Cauchy Differential Evolution Algorithm with Bias Strategy Adaptation Mechanism for Global Numerical Optimization

Tae Jong Choi and Chang Wook Ahn
Department of Computer Engineering, Sungkyunkwan University (SKKU)
2066 Seobu-ro, Suwon 440-746, Republic of Korea
{gry17, cwan}@skku.edu

Abstract—Appropriately adapting mutation strategies is a challenging problem of the literature of the Differential Evolution (DE). The Strategy adaptation Mechanism (SaM) can convert a control parameter adaptation algorithm to a strategy adaptation algorithm. To improve the quality of optimization result, the exploration property is important in the early stage of optimization process and the exploitation property is significant in the late stage of optimization process. To ensure these, we modified the SaM for strictly controlling a balance between the exploration and the exploitation properties, which called the bias SaM (bSaM). We extended the Adaptive Cauchy Differential Evolution (ACDE) by attaching the bSaM. We compared the bSaM with SaM and the bSaM extended ACDE with the state-of-the-art DE algorithms in various benchmark problems. The result of the performance evaluation showed that the bSaM and the bSaM extended ACDE performs better than SaM and the state-of-the-art DE algorithms not only unimodal but also multimodal benchmark problems.

Index Terms—Differential Evolution Algorithm, Adaptive Strategy Control, Adaptive Parameter Control, Exploration Property, Exploitation Property

I. INTRODUCTION

The Differential Evolution (DE) which Storn and Price proposed [1] is population based robust and powerful stochastic search algorithm for continuous space. This algorithm attempts to improve individuals (elements of population) iteratively for finding global optimum(s). This algorithm is simple and contains fewer control parameters than other Evolutionary Algorithms. This advantage attracts many researchers and leads to apply it in many practical problems. The general and detailed information about the DE can be found in [2].

There are three main control parameters in the DE. They are Scaling Factor, Crossover Rate, and Population Size. They have an effect on in optimization performance highly. The result of setting difference between appropriate parameter values and unsuitable parameter values is huge. Although the DE contains few control parameters, the combination of the values of control parameters is infinite. Moreover, there are many mutation strategies which contain different optimization properties. Similar to control parameter case, the result of setting difference between an appropriate mutation strategy and an unsuitable mutation strategy is also huge. Therefore, it is necessary that users should find out the appropriate values of control parameters and mutation strategies in advance. To find out these, users usually utilized the trial-and-error search method. However, this method requires a lot of computational time and space [3-13].

The Strategy adaptation Mechanism (SaM) [8] can convert a control parameter adaptation algorithm to a strategy adaptation algorithm. There are many adaptive parameter control algorithms which show robust and powerful optimization performances. Therefore, by utilizing the SaM, users might easily adapt mutation strategies. The SaM utilizes the strategy probability. The authors of the SaM recommend the SaM with JADE’s [7] adaptive parameter control algorithm for adapting the strategy probability, which performs better than jDE [3] and Uniform distribution adaptive parameter control algorithms.

To improve the quality of optimization result, the exploration property is important in the early stage of optimization process and the exploitation property is significant in the late stage of optimization process [11]. To ensure these properties, we modified the SaM for strictly controlling a balance between the exploration and the exploitation properties, which called the bias SaM (bSaM). We extended the Adaptive Cauchy Differential Evolution (ACDE) [10] by attaching bSaM. The ACDE shows good optimization performance by its adaptive parameter control without adaptive strategy control. We compared the bSaM with SaM and the bSaM extended ACDE with the state-of-the-art DE algorithms in various benchmark problems. The result of the performance evaluation showed that the bSaM is better than the SaM and the ACDE’s adaptive parameter control is better than the JADE’s adaptive parameter control in terms of adapting the strategy probability. In addition, we found out that the Long-tailed distribution still performs better than the Short-tailed distribution for the bSaM.

This work was supported under the framework of international cooperation program managed by National Research Foundation of Korea (NRF-2013K2A1B9066056).
Correspondence should be addressed to Chang Wook Ahn.

© 2014 ACADEMY PUBLISHER
doi:10.4304/jcp.9.9.2139-2145
This paper proceeds as follows. In Section 2, we introduce the standard DE and the ACDE. In Section 3, we describe the proposed algorithm. In Section 4, we present and discuss the result of the performance evaluation. In Section 5, we conclude this paper.

II. RELATED WORK

A. DE Algorithm

The DE [1] is population based stochastic search algorithm for continuous space. Population is a set of \(NP\) individuals. An individual is \(D\)-dimensional vectors, \(x_{i,0} = [x_{i,0}^1, \ldots, x_{i,0}^D]\) represents an individual. Initialization operator initializes population as follows:

\[
x_{i,0} = x_{\text{min}} + \text{rand}[0,1] \cdot (x_{\text{max}} - x_{\text{min}}),
\]

(1)

Here, \(x_{\text{min}} = [x_{\text{min}}^1, \ldots, x_{\text{min}}^D]\) and \(x_{\text{max}} = [x_{\text{max}}^1, \ldots, x_{\text{max}}^D]\) represent the minimum and the maximum bounds of the search space of optimization problem.

There are three main operators in the DE. They are Mutation, Crossover, and Selection. The purpose of Mutation and the Crossover operators is that increase the diversity of population. The purpose of Selection operator is that removes some inferior individuals. The following is one of the mutation operators, called DE/rand/1/bin.

\[
V_{i,j} = x_{i,j} + F \cdot (x_{r_1,j} - x_{r_2,j}),
\]

(2)

Here, \(V_{i,j}\) represents a mutant vector and \(x_{r_1,j}\), \(x_{r_2,j}\), represent some donor vectors. The following is one of the Crossover operators, called Binomial crossover.

\[
u_{i,j} = \begin{cases} 
u_{i,j}^j & \text{if } \text{rand}[0,1] \leq CR \
x_{i,j}^j & \text{otherwise}
\end{cases}
\]

(3)

Here, \(u_{i,j}\) represents a trial vector. The following is Selection operator.

\[
x_{i,j,\text{true}} = \begin{cases} U_{i,j} & \text{if } f(U_{i,j}) \leq f(X_{i,j}) \
x_{i,j} & \text{otherwise}
\end{cases}
\]

(4)

Here, \(f(U_{i,j})\) and \(f(X_{i,j})\) represent the fitness values of the trial vector and the target vector. The DE executes these operators until it satisfying some termination conditions.

A. ACDE

The authors of the ACDE [10] recommend that applying the Long-tailed distribution to the arithmetic mean of a set of successfully evolved individuals’ control parameters at every generations, which can accelerate optimization performance. The followings are the control parameter adaptation of the ACDE.

\[
F_i = C(F_{\text{avg}}, 0.1),
\]

(5)

\[
CR_i = C(CR_{\text{avg}}, 0.1).
\]

(6)

Here, \(C\), \(F_{\text{avg}}\), and \(CR_{\text{avg}}\) represent the Cauchy distribution, the arithmetic mean of a set of successfully evolved individuals’ Scaling Factors, and the arithmetic mean of a set of successfully evolved individuals’ Crossover Rates. The ACDE adapts control parameters appropriately, which leads to robust and powerful optimization performance.

III. THE PROPOSED ALGORITHM

This section describes the strategy adaptation extended ACDE algorithm in detail. The first subsection discusses the strategy pool. The next subsection discusses the strategy adaptation.

A. Strategy Pool

In the proposed algorithm, we applied two mutation strategies. The followings are the strategies.

1) DE/rand/1/bin
2) DE/current-to-best/3/bin

We applied the first mutation strategy for supporting the exploration properties. It performs slow convergence but it has small risk about getting stuck some local minimums. We applied the second mutation strategy for promoting the exploitation properties. It performs fast convergence but it has high probability about getting stuck some local minimums. To increase the number of perturbation vectors can reduce this probability. Therefore, we applied the second mutation strategy instead of DE/current-to-best/1/bin or DE/current-to-best/2/bin.

B. Strategy Adaptation

The SaM can convert a control parameter adaptation algorithm to a strategy adaptation algorithm. \(K\) and \(S_k = \{1, 2, \ldots, K\}\) represent the number of mutation strategies in the strategy pool and the strategy pool. Each individual contains the strategy probability, \(\eta_i \in [0,1)\). The following is the strategy adaptation of the SaM.

\[
S_{i,j} = \eta_i \cdot K + 1.
\]

(7)

The value of the strategy probability determines one of mutation strategies in the strategy pool. For example, if \(K = 2\) and \(\eta_i \in [0,0.5]\), then the first mutation strategy is assigned to the individual \(S_{i,j} = 1\). On the other hand, if \(\eta_i \in [0.5,1]\), then the second mutation strategy is assigned to the individual \(S_{i,j} = 2\). To adapt the strategy probability is similar to adaptive parameter control. Therefore, an adaptive parameter control algorithm can perform as an adaptive strategy control algorithm. However, the SaM has some problems. The followings are the problems of the SaM.

1) The mutation strategies that supporting the exploration property perform slow convergence. Therefore, in the early stage of optimization
TABLE I. BENCHMARK FUNCTIONS

<table>
<thead>
<tr>
<th>Benchmark Functions</th>
<th>Search Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 Sphere Function</td>
<td>[-100, 100]</td>
</tr>
<tr>
<td>F2 Schwefel’s Problem 2.22</td>
<td>[−10, 10]</td>
</tr>
<tr>
<td>F3 Schwefel’s Problem 1.2</td>
<td>[-100, 100]</td>
</tr>
<tr>
<td>F4 Schwefel’s Problem 2.21</td>
<td>[-100, 100]</td>
</tr>
<tr>
<td>F5 Generalized Rosenbrock’s Function</td>
<td>[-30, 30]</td>
</tr>
<tr>
<td>F6 Step Function</td>
<td>[-100, 100]</td>
</tr>
<tr>
<td>F7 Quartic Function with Noise</td>
<td>[-1.28, 1.28]</td>
</tr>
<tr>
<td>F8 Generalized Schwefel’s Problem 2.26</td>
<td>[-500, 500]</td>
</tr>
<tr>
<td>F9 Generalized Rastrigin’s Function</td>
<td>[-5.12, 5.12]</td>
</tr>
<tr>
<td>F10 Ackley’s Function</td>
<td>[-32, 32]</td>
</tr>
<tr>
<td>F11 Generalized Griewank’s Function</td>
<td>[-600, 600]</td>
</tr>
<tr>
<td>F12 Generalized Penalized Functions #1</td>
<td>[-50, 50]</td>
</tr>
<tr>
<td>F13 Generalized Penalized Functions #2</td>
<td>[-50, 50]</td>
</tr>
<tr>
<td>F14 Extended Schaffer</td>
<td>[-100, 100]</td>
</tr>
<tr>
<td>F15 Bohachevsky</td>
<td>[-15, 15]</td>
</tr>
<tr>
<td>F16 Schaffer</td>
<td>[-100, 100]</td>
</tr>
</tbody>
</table>

process, these mutation strategies does not selected with high probability in the SaM, even though these strategies are important in the early stage of optimization process.

2) The mutation strategies that promoting the exploitation property have high probability about getting stuck some local minima. Therefore, these mutation strategies should be selected with high probability after finding out some good regions. However, these are not guaranteed well in the SaM.

To solve these problems, we modified the SaM for strictly controlling a balance between the exploration and the exploitation properties. In the early stage of optimization process, it provides some advantage to the mutation strategies which contain the exploration property, i.e., they can generate trial vectors with higher probability. On the other hand, in the late stage, it offers more chance to the mutation strategies which contain the exploitation property. To modify the SaM, we introduced the bias probability. The following is the modified strategy adaptation of the bSaM.

\[ S_{\alpha} = \left[ \eta_{\alpha} \cdot K + b \right] + 1. \]  

(8)

Here, 0.1 ≤ b ≤ 0.9 is the bias probability. The bias probability increases (or decreases) 0.1 at every certain generations which the maximum generation divided by 10. The sequence of the mutation strategies in the strategy pool determines the operator (increase or decrease) for the bias probability. For example, if the mutation strategies which have the exploration property are assigned to lower numbers and the mutation strategies which have the exploitation property are assigned to higher numbers then, the bias probability starts 0.9 and it decreases 0.1 at every certain generations until it reaches 0.1. On the other hand, if the mutation strategies which have the exploration property are assigned to higher numbers and the mutation strategies which have the exploitation property are assigned to lower numbers then, the bias probability starts 0.1 and it increases 0.1 at every certain generations until it reaches 0.9.

To adapt the strategy probability, we applied the ACDE’s adaptive parameter control. The following is the strategy adaptation of the proposed algorithm.

\[ \eta_{\alpha} = C(\eta_{avg} \cdot 0.1). \]  

(9)

Here, \( \eta_{avg} \) represents the arithmetic mean of a set of successfully evolved individuals’ strategy probabilities.

IV. PERFORMANCE EVALUATION

This section describes the performance evaluation.

A. Experiment Setting

We compared the bSaM with SaM and the bSaM extended ACDE with the state-of-the-art DE algorithms in various benchmark problems. Table I. presents the problems. The following is the compared DE algorithms.

1) ACDE + bSaM with ACDE approach;
2) MDE [13];
3) jDE [3];
4) SaDE [5];
5) Standard DE with F=0.5 and CR=0.9 [1][3];

We fixed the population size at 100 and 400 in dimension 30 and 100 experiments. All values of control parameters used in the performance evaluation were the recommended values by their authors. We executed 100 times independently to gather the results of the performance evaluation, which are the average of optimization results (MEAN) and its standard deviation (STD). To make the result clear, we marked the best result in boldface.

B. Comparison Between the Proposed Algorithm and the State-of-the-Art DE Algorithms

Table II presents the experiment result of comparison in dimension 30 experiments. The result showed that the proposed algorithm found better optimum values than the compared DE algorithms except F3, and F5. In F3, the SaDE showed best optimization performance. In F5, the standard DE showed best optimization performance. The proposed algorithm showed second best optimization performance in F3 and second best optimization performance in F5. Although the proposed algorithm performed not better in some benchmark problems, it outperformed in other benchmark problems.

Table III presents the experiment result of comparison in dimension 100 experiments. The result showed that the proposed algorithm found better optimum values than the compared DE algorithms except F3, F4, F5, F10, F15,
and F16. In F3 and F10, the SaDE showed best optimization performance. In F4, F5, F15, and F16, the jDE showed best optimization performance. The proposed algorithm showed second best optimization performance in F3, F5, F15, and F16 and third best optimization performance in F4 and fifth best optimization performance in F2, F7, F8, and F17.

<table>
<thead>
<tr>
<th>F</th>
<th>D</th>
<th>MAX GEN</th>
<th>F1</th>
<th>1.16E-36</th>
<th>2.05E-36</th>
<th>3.81E-17</th>
<th>3.81E-17</th>
<th>2.20E-28</th>
<th>2.20E-28</th>
<th>2.20E-17</th>
<th>7.24E-20</th>
<th>6.96E-14</th>
<th>4.83E-14</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2</td>
<td>30</td>
<td>2000</td>
<td>1.05E-26</td>
<td>7.55E-27</td>
<td>5.00E-13</td>
<td>1.35E-13</td>
<td>1.35E-13</td>
<td>3.42E-15</td>
<td>3.42E-15</td>
<td>9.55E-10</td>
<td>6.03E-10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F4</td>
<td>30</td>
<td>5000</td>
<td>2.47E-16</td>
<td>3.60E-16</td>
<td>2.12E-08</td>
<td>7.06E-09</td>
<td>9.10E-15</td>
<td>7.50E-14</td>
<td>8.15E-10</td>
<td>2.16E-10</td>
<td>6.95E-02</td>
<td>2.59E-01</td>
<td></td>
</tr>
<tr>
<td>F5</td>
<td>30</td>
<td>20000</td>
<td>1.99E-01</td>
<td>8.69E-01</td>
<td>1.20E-01</td>
<td>6.80E-01</td>
<td>1.20E-01</td>
<td>6.80E-01</td>
<td>1.33E+01</td>
<td>1.15E+01</td>
<td>1.97E-30</td>
<td>9.13E-30</td>
<td></td>
</tr>
</tbody>
</table>

**TABLE II.**


| F  | D  | MAX GEN | MDE | 7.54E-17 | 3.81E-17 | 1.89E-28 | 2.20E-28 | 3.36E-20 | 7.24E-20 | 6.96E-14 | 4.83E-14 |
|----|----|---------|-----|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| F2 | 30 | 2000    | jDE | 5.00E-13 | 1.35E-13 | 1.65E-23 | 1.31E-23 | 6.31E-15 | 3.42E-15 | 9.55E-10 | 6.03E-10 |
| F3 | 30 | 5000    | SaDE | 3.67E-01 | 6.38E-02 | 2.78E-14 | 4.55E-14 | 2.37E-35 | 8.78E-35 | 3.69E-11 | 5.32E-11 |
| F4 | 30 | 5000    | 2.12E-08 | 7.06E-09 | 9.10E-15 | 7.50E-14 | 8.15E-10 | 2.16E-10 | 6.95E-02 | 2.59E-01 |
| F5 | 30 | 20000   | 1.20E-01 | 6.80E-01 | 1.20E-01 | 6.80E-01 | 1.33E+01 | 1.15E+01 | 1.97E-30 | 9.13E-30 |

**TABLE III.**

**COMPARISON BETWEEN THE PROPOSED ALGORITHM AND THE STATE-OF-THE-ART DE ALGORITHMS (DIMENSION = 100)**

| F  | D  | MAX GEN | MDE | 7.54E-17 | 3.81E-17 | 1.89E-28 | 2.20E-28 | 3.36E-20 | 7.24E-20 | 6.96E-14 | 4.83E-14 |
|----|----|---------|-----|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| F2 | 30 | 2000    | jDE | 5.00E-13 | 1.35E-13 | 1.65E-23 | 1.31E-23 | 6.31E-15 | 3.42E-15 | 9.55E-10 | 6.03E-10 |
| F3 | 30 | 5000    | SaDE | 3.67E-01 | 6.38E-02 | 2.78E-14 | 4.55E-14 | 2.37E-35 | 8.78E-35 | 3.69E-11 | 5.32E-11 |
| F4 | 30 | 5000    | 2.12E-08 | 7.06E-09 | 9.10E-15 | 7.50E-14 | 8.15E-10 | 2.16E-10 | 6.95E-02 | 2.59E-01 |
| F5 | 30 | 20000   | 1.20E-01 | 6.80E-01 | 1.20E-01 | 6.80E-01 | 1.33E+01 | 1.15E+01 | 1.97E-30 | 9.13E-30 |

and F16. In F3 and F10, the SaDE showed best optimization performance. In F4, F5, F15, and F16, the jDE showed best optimization performance. The proposed algorithm showed second best optimization performance in F3, F5, F15, and F16 and third best optimization performance in F4 and fifth best optimization performance in F4 and fifth best...
optimization performance in F10. Similar to the dimension 30 experiments, although the proposed
algorithm performed not better in some benchmark problems, it outperformed in other benchmark problems. C. Comparison Between bSaM with ACDE Parameter Adaptation and the SaM with JADE Parameter Adaptation

We attempted to verify that the extended ACDE by the bSaM with ACDE approach is more useful than the extended ACDE by the SaM with JADE approach. Table IV presents the experiment result of comparison. The result showed that the extended ACDE by the bSaM with ACDE approach found better optimum values than the extended ACDE by the SaM with JADE approach except F3 and F4. However, in F1, F2, F10, F11, F14, F15, and F16, the extended ACDE by the bSaM with ACDE approach found much better optimum values. As a result, the extended ACDE by the bSaM with ACDE approach can find better optimum values than the extended ACDE by the SaM with JADE approach in various problems.

D. Comparison Between bSaM with ACDE Parameter Adaptation and the SaM with ACDE Parameter Adaptation

We attempted to verify that the extended ACDE by the bSaM with ACDE approach is more useful than the extended ACDE by the SaM with ACDE approach. Table V presents the experiment result of comparison. The result showed that the extended ACDE by the bSaM with ACDE approach found better optimum values than the extended ACDE by the SaM with ACDE approach except F3, F4, F13, and F14. However, the differences of F13, F14 were not significant and in F10, F11, F15 and F16, the extended ACDE by the bSaM with ACDE approach algorithm found much better optimum values. As a result, the extended ACDE by the bSaM with ACDE approach can find better optimum values than the extended ACDE by the SaM with ACDE approach in various problems.

E. Comparison Between bSaM with ACDE Parameter Adaptation and the bSaM with JADE Parameter Adaptation

We attempted to verify that the extended ACDE by the bSaM with ACDE approach is more useful than the extended ACDE by the bSaM with JADE approach. Table VI presents the experiment result of comparison. The result showed that the extended ACDE by the bSaM with ACDE approach found better optimum values than the extended ACDE by the bSaM with JADE approach except F5. However, in F1, F3, F4, F14, and F16, the extended ACDE by the bSaM with ACDE approach found much better optimum values. As a result, the extended ACDE by the bSaM with ACDE approach can find better optimum values than the extended ACDE by the bSaM with JADE approach in various problems.

F. Comparison Between the Long-tailed and the Short-tailed Distributions for the bSaM

We attempted to verify that the Lang-tailed distribution is more useful than the Short-tailed distribution in terms of the strategy adaptation of the bSaM with ACDE approach. Table VII presents the experiment result of comparison.
comparison. We applied the Cauchy and Gaussian distributions as the Long-tailed and the Short-tailed distributions to the strategy adaptation of the bSaM. The result showed that the Long-tailed distribution found better optimum values than the Short-tailed distribution except F3. However, in F1, F2, F4, F14, F15, and F16, the Long-tailed distribution found much better optimum values. As a result, the Long-tailed distribution performs better than the Short-tailed distribution for adapting the strategy parameter of the bSaM.

G. Comparison Between the Extended ACDE by bSaM with ACDE Parameter Adaptation with the ACDE

Table VIII presents the experiment result of comparison in dimension 30 experiments. The result showed that the extended ACDE by bSaM with ACDE approach found better optimum values than the ACDE except F2, F4, F10, F14, F15, and F16. However, the differences of F10 and F13 were not significant and in F3, F4, and F5, the extended ACDE by the bSaM with ACDE approach found much better optimum values.

Table IX presents the experiment result of comparison in dimension 100 experiments. The result showed that the extended ACDE by bSaM with ACDE approach found better optimum values than the ACDE except F2, F4, F10, F15, and F16. However, the difference of F2 was not significant and in F1, F3, F5, F7, F12, and F13, the extended ACDE by the bSaM with ACDE approach found much better optimum values.

As a result, the extended ACDE by the bSaM with ACDE approach can find better optimum values than the ACDE in various problems.

V. CONCLUSION

In this paper, we modified the SaM for strictly controlling a balance between the exploration and the exploitation properties, which called the bSaM. We extend the ACDE to attach the bSaM. The performance evaluation showed that the bSaM performed better than the SaM and the ACDE’s adaptive parameter control performed better than the JADE’s adaptive parameter control in terms of adapting the strategy probability. In addition, we found out that the Long-tailed distribution still performs better than the Short-tailed distribution for the bSaM.

REFERENCES


Tae Jong Choi received M.S at the Department of Computer Engineering, Ajou University, Republic of Korea. He is currently working toward Ph.D degree at the Department of Computer Engineering, Sungkyunkwan University (SKKU), Republic of Korea. His research interests include genetic algorithms and the applications of evolutionary techniques to wireless networks.

Chang Wook Ahn received Ph.D degree at the Department of Information and Communications, Gwang-Ju Institute of Science & Technology (GIST), Republic of Korea. From 2007 to 2008, he was a Research Professor in GIST. He is currently an Assistant Professor in Sungkyunkwan University (SKKU), Republic of Korea. His research interests include genetic algorithms, multiobjective optimization, neural networks, and the applications of evolutionary techniques to wireless networks.