An Adaptive Differential Evolution Algorithm

Nasimul Noman, Danushka Bollegala and Hitoshi Iba
Graduate School of Engineering
University of Tokyo
Tokyo 113-8656, Japan
Email: {noman, danushka, iba}@iba.t.u-tokyo.ac.jp

***Abstract*—The performance of Differential Evolution (DE) algorithm is significantly affected by its parameter setting. But the choice of parameters is heavily dependent on the problem characteristics. Therefore, recently a couple of adaptation schemes that automatically adjust DE parameters have been proposed. The current work presents another adaptation scheme for DE parameters namely amplification factor and crossover rate. We systematically analyze the effectiveness of the proposed adaptation scheme for DE parameters using a standard benchmark suite consisting of ten functions. The undertaken empirical study shows that the proposed adaptive DE (aDE) algorithm exhibits an overall better performance compared to other prominent adaptive DE algorithms as well as canonical DE.**

I. Introduction

Differential Evolution (DE), proposed by Storn and Price in 1995 [1], is considered as one of the most reliable and versatile optimization techniques available today. DE is a parallel stochastic search technique designed within the common framework of Evolutionary Algorithm (EA). The effectiveness and superiority of the algorithm in solving a wide range of optimization problems have been proven in many studies.

DE is a population-based optimizer that works utilizing the concepts borrowed from EAs [2]. The algorithm starts by sampling the search space at multiple, randomly selected search points. Like other EAs, DE creates new search points through perturbations of the existing points. Using a differential mutation and a recombination operaton DE creates new search points which are evaluated against their parents. Then a knock-out selection mechanism is applied that deterministically promotes the winners to the next generation. And this cycle of perturbation and selection is iterated generation after generation until the termination criteria is satisfied [3].

Since its introduction, DE has been applied to solve problems in many scientific and engineering fields. Application of DE ranges from acoustics to aerodynamics, biotechnology to chemical engineering, environmental sciences to electrical engineering, industry to economics, medicine to transportation [4]. Such wide range of applications of the algorithm is sufficient to justify its effectiveness and versatility.

Besides its robust and reliable performance, another reason behind the popularity of DE is its simple and easy-to-understand structure. The canonical DE needs setting of only three parameters: amplification factor (F), crossover rate (CR) and population size (P). Although DE works with a handful of parameters, its performance is very sensitive to these parameters, particularly to F and CR [5]. Several studies have shown that improper setting of these parameters may cause DE to perform really poor [6], [7]. Although, there is some guideline available on DE parameter settings [8], selection of the best parameter set is a problem dependent task which is done using either the trial and error method or the grid search.

In recent years a couple of methods have been proposed to adjust F and CR automatically. Liu and Lampinen have proposed Fuzzy Adaptive Differential Evolution (FADE) that employs fuzzy logic controllers to adapt F and CR parameters of DE [5]. Qin and Suganthan suggested to select DE's learning scheme, as well as parameter setting, in response to the learning experience in their proposed Self-adaptive DE (SaDE) algorithm [9].

Brest *et al.* [10] proposed a self-adaptation method for F and CR in their proposed jDE algorithm. Das *et al.* [11] have proposed two variants of DE, DERSF, and DETVSF, using varying scale factors. In a couple of recent work, the chaos theory has been successfully applied to automatically adjust the parameters of DE [12], [13], [14].

However, almost all the adaptation schemes available today actually apply some sort of random change to the parameters irrespective the quality of the current parameters. Therefore, in this work we present another adaptation scheme that preserves better parameter choices and changes the parameter values that are not being productive. We implemented this adaptation scheme at individual level and the modified DE version is called aDE.

We evaluated the performance of the new variant of DE using a test suite of ten well-known benchmark functions. Error in the final solution, required fitness evaluation to locate the pseudo-global optimum and the convergence trace were taken into consideration in our evaluation. We compared the performance of aDE with canonical DE and two other prominent variants of DE namely jDE and chaotic DE (chDE). The numerical study shows that in general aDE outperformed the canonical DE and other adaptive variants of DE. The results point out that the proposed scheme is more effective in DE parameter adaptation compared to the existing ones.

The rest of the paper is organized in the following manner. Section II presents the canonical Differential Evolution (DE) algorithm. Two adaptive versions of DE namely jDE and chDE are described in Section III. The newly proposed adaptive DE (aDE) algorithm is presented in Section IV. The empirical study on canonical and adaptive DE algorithms, results and analysis are presented in Section V. Finally, Section VI concludes the paper with a summary of the study.

II. Differential Evolution

Storn and Price proposed Differential Evolution (DE) as a family of algorithms to solve real-parameter optimization problems [1]. The variants of DE are differentiated from each other by varying the mutation and/or the recombination operation within a common framework. However, in this study we used the DE/rand/1/exp variant of DE. Therefore, we describe DE with this variant.

DE works with a population of individuals $\boldsymbol{x}_G^i$, $i = 1, 2, \cdots, P$ each representing a solution to the problem. DE individuals are encoded as real vectors of size N which is the dimension of the problem. The number of individuals in a population is called population size and is denoted by P and the generation number is denoted by G. The initial population, $\mathcal{P}_1$ is created by randomly creating the vectors in appropriate search ranges. Then the fitness score of each individual is calculated through evaluation.

DE practices random parent selection regardless of their fitness values. In every generation, each individual $\boldsymbol{x}_G^i$ gets a chance to become the principal parent and to breed its own offspring mating with other randomly chosen auxiliary parents. Formally, for every principal parent $\boldsymbol{x}_G^i$, $i = 1, 2, \cdots, P$, three other auxiliary parents $\boldsymbol{x}_G^{r1}$, $\boldsymbol{x}_G^{r2}$, $\boldsymbol{x}_G^{r3}$ are selected randomly such that $r1, r2, r3 \in \{1, 2, \cdots, P\}$ and $i \neq r1 \neq r2 \neq r3$. Then these three auxiliary parents participate in *differential mutation* operation to create a mutated individual $\boldsymbol{x}_G^{mut}$ as follows:

$$\boldsymbol{x}_G^{mut} = \boldsymbol{x}_G^{r1} + F(\boldsymbol{x}_G^{r2} - \boldsymbol{x}_G^{r3}) \tag{1}$$

where F is the amplification factor, a real-valued control parameter chosen from [0.1, 1.0] [3]. Subsequently, the mutated vector, $\boldsymbol{x}_G^{mut}$, participates in exponential crossover operation with the principal parent $\boldsymbol{x}_G^i$ to generate the trial individual or offspring $\boldsymbol{x}_G^{child}$. Exponential crossover is actually a cyclic two-point crossover in which CR, another control parameter of DE, determines how many consecutive genes of the mutated vector, $\boldsymbol{x}_G^{mut}$, on average are copied to the offspring $\boldsymbol{x}_G^{child}$.

The selection scheme used in DE is also known as *knock-out competition*. As the name suggests, DE plays a one-to-one competition between the principal parent, $\boldsymbol{x}_G^i$, and its offspring $\boldsymbol{x}_G^{child}$ to select the survivor for the next generation. The DE selection scheme can be described as follows:

$$\boldsymbol{x}_{G+1}^i = \begin{cases} \boldsymbol{x}_G^{child} & \text{if } f(\boldsymbol{x}_G^{child}) \text{ is better than } f(\boldsymbol{x}_G^i) \\ \boldsymbol{x}_G^i & \text{otherwise} \end{cases} \tag{2}$$

Repeating the above mentioned mutation and crossover operations on each individual of the current generation, DE creates a new generation of population which replaces the current generation. And this generation alternation process is iterated until the termination criteria is satisfied. The control parameters of DE (F,CR and P) are chosen beforehand and are kept constant throughout the search in this canonical version of the algorithm. The pseudo-code description of canonical DE is presented in Algorithm 1.

Algorithm 1 DE

```
1: Select $P$, $F$ and $CR$ and Set $G = 1$
2: $\mathcal{P}_G$ = initialize population randomly
3: while termination criteria not satisfied do
4:     for each individual $\boldsymbol{x}_G^i$ in $\mathcal{P}_G$ do
5:         Select auxiliary parents $\boldsymbol{x}_G^{r1}$, $\boldsymbol{x}_G^{r2}$ and $\boldsymbol{x}_G^{r3}$
6:         Create offspring $\boldsymbol{x}_G^{child}$ using mutation and crossover
7:         $\mathcal{P}_{G+1}$ = $\mathcal{P}_{G+1} \cup$ Best($\boldsymbol{x}_G^{child}$,$\boldsymbol{x}_G^i$)
8:     end for
9:     Set $G = G + 1$
10: end while
```

III. Related Studies

Among the three control parameters of DE, the algorithm is most sensitive to the setting of F and CR. Therefore, several attempts have been made to automatically adjust these parameters based on the problem characteristics. A brief overview of these methods have been presented in Section I. In this section we describe two most prominent parameter adaptation schemes for DE namely jDE and chaotic DE (chDE) which will be empirically compared with the newly proposed aDE algorithm.

A. jDE

Brest *et al.* [10] proposed the jDE algorithm that uses a self-adaptive mechanism to adjust the control parameters F and CR during the evolutionary process. In order to encode these two control parameters in individual vectors, the size of each individual, $\boldsymbol{x}_G^i$, is extended by two. In other words, each individual has its own copies of F and CR which are encoded in it and evolved with it. Therefore the size of each individual in jDE algorithm is $(N + 2)$ where N is the size of the objective vector.

For evolving new generation of individuals, the original DE mutation and crossover operations are applied. However, these genetic operations work only on the objective vector part of the individual. The F and CR for new offspring are calculated as follows:

$$F_{G+1}^i = \begin{cases} F_l + rand_1 * F_u, & \text{if } rand_2 < \tau_1, \\ F_G^i & \text{otherwise} \end{cases} \tag{3}$$

$$CR_{G+1}^i = \begin{cases} rand_3 & \text{if } rand_4 < \tau_2, \\ CR_G^i & \text{otherwise} \end{cases} \tag{4}$$

where $rand_j$, $j \in \{1, 2, 3, 4\}$, are uniform pseudo-random values $\in$ [0, 1], τ_i and τ_2 are constants that represent the probabilities to adjust F and CR respectively. F_l represents the minimum value that can be assigned to F_{G+1}^i and the F_u represents the maximum random variation added to F_l. It should be noted that F_i^{G+1} and CR_{G+1}^i are obtained before the mutation is performed. So they influence the mutation, crossover, and selection operations of the new individual

$\boldsymbol{x}_G^{child}$ [10]. In this study, $F_l = 0.1$, $F_u = 0.2$ and $\tau_1 = \tau_2 = 0.1$ were used as suggested in the original proposal [10].

So there are two differences between jDE and the canonical DE. The first one is the realization of the F and CR parameters at individual level and the other one is the introduction of additional rules for calculating F and CR at individual level.

B. Chaotic DE

Chaos describes the complex behavior of a nonlinear deterministic system which is dynamic, pseudo-random, ergodic and sensitive to its initial condition [15]. Recently the chaotic systems have been used instead of random processes in different fields including the optimization theory. Different types of chaotic equations have been considered in literature such as Logistic map, Lozi map, Lorenz system, Tent map etc. [13].

As described earlier, many hybrids of DE have been proposed that incorporates chaotic systems in many different ways. The most common application of chaos theory in DE is for its parameter adaptation [13], [12], [16], [14]. Besides chaotic sequences have been utilized to initialize DE population [17], to re-initialize the converging population or to increase population diversity [13], to perform local search in the neighborhood of the selected individuals [18] etc. And in these tasks many different types of chaotic systems have been applied. Study of all these techniques is clearly out of the scope of this work. Therefore, in current study we used the most commonly used chaotic sequence, known as logistic map, to adapt the DE control parameters. The logistic equation is defined as follows:

$$y(k) = \mu \cdot y(k-1) \cdot [1 - y(k-1)] \tag{5}$$

where k is the sample, and μ is a control parameter, $0 \leq \mu \leq 4$. The value of μ determines whether y stabilizes at a constant size, oscillates between a limited sequence, or behaves chaotically [16]. The logistic equation is deterministic, and exhibits chaotic dynamics when $\mu = 4$ and $y(1) \notin \{0.00, 0.25, 0.50, 0.75, 1.00\}$. In this work we used $\mu = 4$ and $y(1) \notin \{0.00, 0.25, 0.50, 0.75, 1.00\}$ so that y is in the range $[0, 1]$.

Here, in the chaotic version of DE (chDE), we initialized and adjusted the F and CR parameters of DE using the logistic independent sequences as described before. However, the same F and CR values were used population-wide which were updated as follows:

$$F_{G+1} = \mu \cdot F_G * [1 - F_G] \tag{6}$$

$$CR_{G+1} = \mu \cdot CR_G * [1 - CR_G] \tag{7}$$

Here, F and CR were initialized randomly satisfying the constraint mentioned earlier. All the other characteristics of DE were retained intact in this chDE algorithm.

IV. Proposed Adaptive DE

This section presents our newly proposed adaptation scheme for DE parameters. As discussed before, most of the adaptation schemes for DE parameter use some sort of random adjustment of the parameters. We believe that random adjustment can be a good choice only when parameter setting is not suitable. But random adjustment of parameters regardless of their effectiveness may cause to lose high-quality parameter values. Therefore, in the proposed adaptive DE (aDE) algorithm we preserve the parameter settings which produce high quality individuals and adjust the parameter values in other cases.

Now the question is how to decide that the current parameter setting is effective? In the current implementation we adopted a pretty straightforward strategy. We compared the fitness of the offspring ($f(\boldsymbol{x}_G^{child})$) with the average fitness value of the current generation (f_{avg}). If $f(\boldsymbol{x}_G^{child})$ is better than f_{avg} then we retain the F and CR of the principal parent $\boldsymbol{x}_G^i$ in offspring otherwise we change them randomly. Formally, the choice of F and CR in offspring $\boldsymbol{x}_G^{child}$ is done as follows:

$$F_G^{child} = \begin{cases} F_G^i & \text{if } f(\boldsymbol{x}_G^{child}) < f_{avg}, \\ Rand(0.1, 1.0) & \text{otherwise} \end{cases} \tag{8}$$

$$CR_G^{child} = \begin{cases} CR_G^i & \text{if } f(\boldsymbol{x}_G^{child}) < f_{avg}, \\ Rand(0.0, 1.0) & \text{otherwise} \end{cases} \tag{9}$$

where $f(\cdot)$ is a minimization problem and $Rand(m, n)$ returns a uniform random number between m and n.

From the above description it is evident that the adaptation scheme is implemented at individual level as in jDE. We augmented the objective vector of each individual with its own F and CR parameters. Initially the F and CR values are created randomly in each individual. Whenever a new individual is created its F and CR values are chosen according to the above described scheme. However, the objective vector part of the offspring is created using the original mutation and crossover operations of DE.

Algorithm 2 aDE

```
1:  Select $P$ and Set $G = 1$
2:  for each individual $x_G^i$ do
3:      initialize the object vector randomly
4:      set $F_G^i = Rand(0.1, 1.0)$
5:      set $CR_G^i = Rand(0.0, 1.0)$
6:  end for
7:  while termination criteria not satisfied do
8:      for each individual $x_G^i$ in $\mathcal{P}_G$ do
9:          Select auxiliary parents $x_G^{r1}$, $x_G^{r2}$ and $x_G^{r3}$
10:         Create offspring $x_G^{child}$ using mutation and crossover
11:         Set $F_G^{child}$ using Eq. (8)
12:         Set $CR_G^{child}$ using Eq. (9)
13:         $\mathcal{P}_{G+1}$ = $\mathcal{P}_{G+1} \cup$ Best($x_G^{child}$,$x_G^i$)
14:     end for
15:     Set $G = G + 1$
16: end while
```

The pseudo-code description of aDE is presented in Algorithm 2. Other than the adaptation scheme there is another difference between jDE and the newly proposed aDE algorithm. In jDE the newly adapted F and CR values are used to create new offspring. On the other hand, in aDE the F and CR of the principal parent x_G^i is used to create new offspring and then

TABLE I
BENCHMARK SUITE

Sphere	$f_{sph}(\vec{\mathbf{x}}) = \sum_{i=1}^{N} x_i^2$	$[-100, 100]^{30}$
Elliptic	$f_{ell}(\vec{\mathbf{x}}) = \sum_{i=1}^{N} \left(10^6\right)^{\left(\frac{i-1}{N-1}\right)} x_i^2$	$[-100, 100]^{30}$
Schwefel 1.2	$f_{sch}(\vec{\mathbf{x}}) = \sum_{i=1}^{N} \left(\sum_{j=1}^{i} x_j\right)^2$	$[-100, 100]^{30}$
Ackley	$f_{ack}(\vec{\mathbf{x}}) = 20 + e - 20e^{\left(-0.2\sqrt{\left(\frac{1}{N}\sum_{i=1}^{N} x_i^2\right)}\right)} - e^{\left(\frac{1}{N}\sum_{i=1}^{N} \cos(2\pi x_i)\right)}$	$[-32, 32]^{30}$
Rastrigin	$f_{ras}(\vec{\mathbf{x}}) = 10N + \sum_{i=1}^{N}(x_i^2 - 10\cos(2\pi x_i))$	$[-5.12, 5.12]^{30}$
Griewank	$f_{grw}(\vec{\mathbf{x}}) = \sum_{i=1}^{N} x_i^2/4000 - \prod_{i=1}^{N} \cos(x_i/\sqrt{i}) + 1$	$[-600, 600]^{30}$
Rosenbrock	$f_{ros}(\vec{\mathbf{x}}) = \sum_{i=1}^{N-1}(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2)$	$[-100, 100]^{30}$
Weierstrass	$f_{wrs}(\vec{\mathbf{x}}) = \sum_{i=1}^{N}\left(\sum_{k=0}^{k_{max}} (a^k \cos(2\pi b^k(x_i + 0.5)))\right) - N\sum_{k=0}^{k_{max}} (a^k \cos(\pi b^k)$	$[-0.5, 0.5]^{30}$
Schaffer	$f_{scf}(\vec{\mathbf{x}}) = \sum_{i=1}^{N} F(x_i, x_{i+1}); \qquad x_{N+1} = x_1$ where, $F(x, y) = 0.5 + \frac{sin^2(\sqrt{x^2+y^2}) - 0.5}{(1+0.001(x^2+y^2))^2}$	$[-0.5, 0.5]^{30}$
Salomon	$f_{sal}(\vec{\mathbf{x}}) = 1 - \cos(2\pi \Vert\mathbf{x}\Vert) + 0.1\Vert\mathbf{x}\Vert$ where, $\Vert\mathbf{x}\Vert = \sqrt{\left(\sum_{i=1}^{N} x_i^2\right)}$	$[-100, 100]^{30}$

its parameters are chosen adaptively. From the description of Algorithm 2 and the adaptation scheme it is evident that the newly proposed aDE does not increase the time complexity in comparison to the canonical DE or other adaptive DEs.

V. EMPIRICAL STUDY

In this section we evaluate the performance of the newly proposed aDE algorithm using numerical experiment. We point out the strength and weakness of the adaptation scheme evaluating it using a test suite of ten benchmark functions. The benchmark suite, tabulated in Table I, consists of functions that differs in terms of various characteristics such as the number of local and global minima, symmetry, level of epistasis etc. All these problems were studied at $N = 30$ dimension within the search range shown in Table I.

In order to justify the competitiveness of the proposed aDE algorithm we compared it with canonical DE and two other adaptive variants of DE described in Section III. All the studied algorithms (DE, jDE, chDE and aDE) were implemented using the DE/rand/1/exp strategy of differential evolution.

A. Experimental Setup

For each benchmark we performed 50 independent trial runs on each algorithm starting from different random initial solutions. However, to make the comparison even more unbiased we used the same set of random population to initialize all four algorithms in different trial runs. In each trial, an algorithm was allowed $N * 10,000$ fitness evaluations at maximum to solve the problem. The best *error value* reached by an algorithm at the end of optimization was recorded. The *error value* for a solution x is defined as $(f(x) - f(x^*))$ where x^* denotes the optimum solution for $f(\cdot)$. The average and standard deviation of these error values over 50 trial runs are reported in Table II. We also kept track of that if any algorithm could reach an error value $< 10^{-8}$ (we call it pseudo global optimum) in any of these trial runs. We recorded the number of fitness evaluations required to reach the pseudo global optimum and tabulated these results in Table III along with the number of trials in which pseudo global optimum was reached. The best results are shown in boldface in these tables.

For all studied algorithms the same population size $P = 100$ was used. In DE the other parameters were set as: amplification factor $F = 0.5$ and crossover probability $CR = 0.9$. chDE, jDE and the proposed aDE do not require any additional parameter setting. In jDE and aDE $F_{G=1}^{i}$ and $CR_{G=1}^{i}$ were randomly initialized from $[0.1, 1.0]$ and $[0, 1]$ respectively. In chDE F and CR were initialized randomly from $[0, 1]$ such that they are not in $\{0.00, 0.25, 0.50, 0.75, 1.00\}$.

B. Results

f_{sph} and f_{ell} are unimodal, symmetric and separable functions. Therefore, for all the variants of DE it was easy to locate the (pseudo) global optimum very easily in every trial. However, in terms of the acquired accuracy level there was certain differences among these algorithms. The best error values were achieved by aDE. jDE and chDE exhibited similar performance which was worse than aDE but better than canonical DE (Table II). In terms of required fitness evaluation (Table III), aDE performed best, chDE and jDE did similar and DE performed worst. The convergence curves of Fig 1 and 2 also ascertain the speedup achieved by the newly proposed aDE. chDE and jDE improved the convergence characteristics of DE but aDE has outperformed them.

TABLE II
PERFORMANCE COMPARISON OF DE, jDE, chDE AND aDE IN TERMS OF ERROR VALUE (FITNESS)

Fun	DE	jDE	chDE	aDE
f_{sph}	5.45E-37 ± 3.05E-37	4.68E-39 ± 2.86E-39	1.95E-39 ± 3.30E-39	**4.66E-57 ± 6.84E-57**
f_{ell}	1.68E-33 ± 8.70E-34	2.00E-35 ± 1.34E-35	8.16E-36 ± 1.22E-35	**1.05E-52 ± 4.46E-52**
f_{sch}	4.26E-04 ± 1.34E-04	2.83E-02 ± 4.20E-02	4.73E-07 ± 6.64E-07	**5.27E-16 ± 1.43E-15**
f_{ack}	4.23E-15 ± 1.78E-15	3.80E-15 ± 1.67E-15	**3.38E-15 ± 1.44E-15**	3.52E-15 ± 1.53E-15
f_{ras}	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_{grw}	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_{ros}	1.28E+00 ± 8.89E-01	2.47E+01 ± 1.59E+01	1.04E+01 ± 1.09E+01	**3.78E-01 ± 1.58E+00**
f_{wrs}	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_{scf}	1.67E+00 ± 1.44E-01	5.97E-01 ± 7.85E-02	**3.03E-01 ± 4.12E-02**	6.15E-01 ± 7.80E-02
f_{sal}	**2.04E-01 ± 1.98E-02**	3.68E-01 ± 4.68E-02	2.94E-01 ± 5.86E-02	2.06E-01 ± 2.40E-02

TABLE III
PERFORMANCE COMPARISON OF DE, jDE, chDE AND aDE IN TERMS OF FITNESS EVALUATION

Fun	DE	jDE	chDE	aDE
f_{sph}	93281.3 ± 971.6 (50)	89140.2 ± 1111.9 (50)	88064.2 ± 2544.0 (50)	**69297.5 ± 1860.5 (50)**
f_{ell}	118676.0 ± 1126.2(50)	114346.2 ± 1224.8(50)	113121.1 ± 2324.9(50)	**87815.2 ± 2055.0 (50)**
f_{sch}	—	—	—	**194024.0 ± 9721.2 (50)**
f_{ack}	144054.0 ± 1193.4(50)	139252.5 ± 1558.4(50)	134870.5 ± 3598.6(50)	**108243.9 ± 2635.5 (50)**
f_{ras}	219437.2 ± 3722.9(50)	112621.0 ± 1399.8(50)	**98825.7 ± 3188.0 (50)**	110384.6 ± 2765.9 (50)
f_{grw}	99369.2 ± 3262.1 (50)	96841.4 ± 3993.0 (50)	91439.3 ± 2902.2 (50)	**76072.6 ± 3426.6 (50)**
f_{ros}	—	—	—	**286136.0 ± 10346.4 (2)**
f_{wrs}	168788.3 ± 1477.3(50)	147376.8 ± 1526.7(50)	136288.1 ± 3029.1(50)	**119190.3 ± 2254.3 (50)**
f_{scf}	—	—	—	—
f_{sal}	—	—	—	—

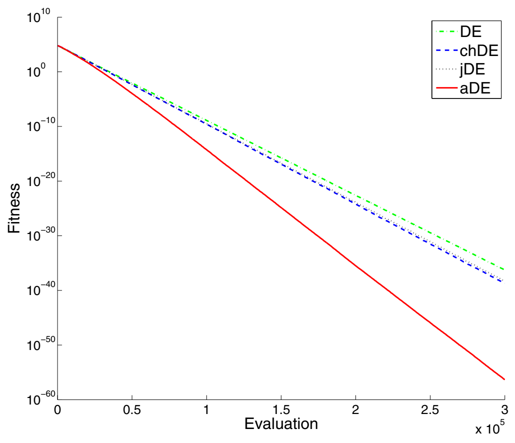

Fig. 1. Search traces for f_{sph} function

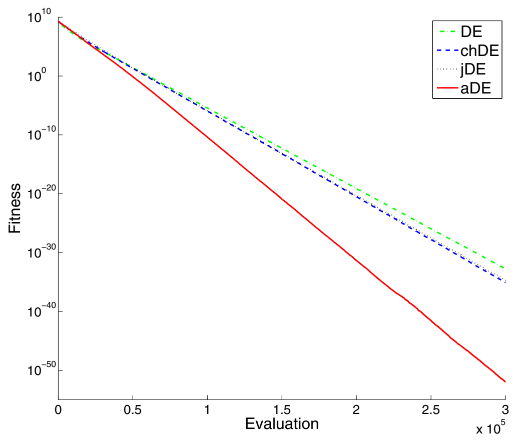

Fig. 2. Search traces for f_{ell} function

f_{sch}, the other unimodal function, is asymmetric and partially separable. In this difficult unimodal problem, none of canonical DE, jDE and chDE could reach an error level $< 10^{-8}$. Only the proposed aDE algorithm could locate the global optimum in every trial. Obviously in terms of error value, as shown in Table II, the best performance was exhibited by aDE. On the other hand, compared to canonical DE, the performance of chDE and jDE was better and poor, respectively. The convergence graph of Fig. 3 shows that although aDE started with a convergence curve similar to DE, through adaptation it improved the convergence characteristics of the algorithm significantly.

The Ackley function (f_{ack}) is a non-separable, symmetric and multimodal function. In terms of error values, as shown in Table II, the best performing algorithm was chDE. But there is no significant difference between aDE and chDE in terms of error value (Table IV). However, in terms of fitness evaluations (Table III) the performance difference is very apparent and significant. aDE took fewest fitness evaluations to locate the pseudo global optimum. Fig. 4 also ascertain that the best convergence characteristic is displayed by aDE algorithm.

f_{ras} is a separable and symmetric multimodal function. Therefore, all the variants of DE could locate the absolute

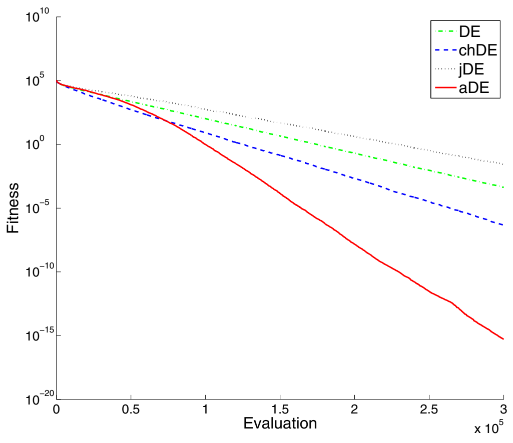

Fig. 3. Search traces for f_{sch} function

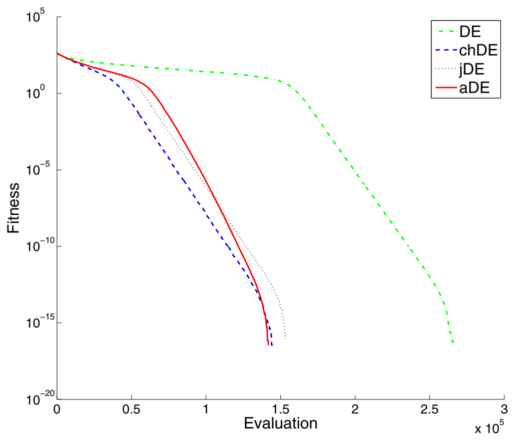

Fig. 5. Search traces for f_{ras} function

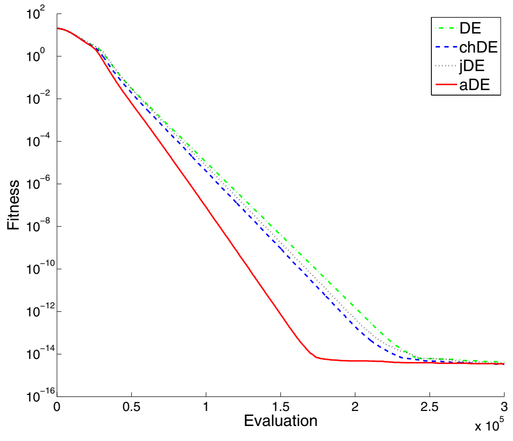

Fig. 4. Search traces for f_{ack} function

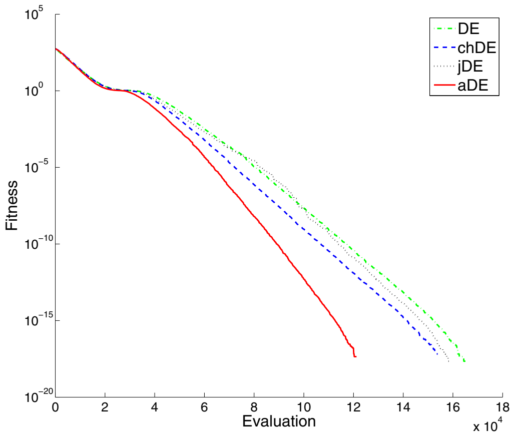

Fig. 6. Search traces for f_{grw} function

global optimum in every trial as shown in Table II. However, in terms of fitness evaluation (to locate the pseudo global optimum) the best performance was exhibited by chDE (Table III) and its performance was significantly better than that of aDE (Table IV). However, if we look at convergence curve of Fig. 5, we learn that initially the convergence characteristics of aDE was poor compared to chDE and jDE but later it improved dramatically and eventually finished fastest. Nevertheless, since we recorded fitness evaluation when error $< 10^{-8}$, chDE became the best performing algorithm for f_{ras} in Table III.

The Griewang function (f_{grw}) is an asymmetric, non-separable and multimodal problem. All DE variants converged to the absolute global optimum in every trial run of f_{grw} (Table II). However, Table III discloses that the proposed aDE algorithm reached the global optimum using fewest fitness evaluations. The search traces of Fig. 6 also show that the parameter adaptation scheme in aDE has improved the convergence characteristics of the classic DE algorithm most significantly.

The Rosenbrock (f_{ros}), an asymmetric multimodal function, is well known for the strong epistasis among its parameters. Because of its high epistatic characteristics, it becomes very challenging to many optimization algorithm to locate the global optimum of f_{ros}. And in this study, only aDE was successful to locate the pseudo global optimum in only two trials out of 50 (Table III). As shown in Table II, the best error value was achieved by aDE and the canonical DE showed better performance compared to jDE and chDE. The convergence graph of Fig. 7 show that aDE reached the plateau of the function at first. But after reaching the plateau all algorithms remained stuck there for a while. However, aDE also managed to exit the plateau first and start to progress towards global optimum. So it seems that the adaptation scheme helped the algorithm to adjust its parameters effectively once again.

The Weierstrass (f_{wrs}) is a non-separable multimodal function. But all the DE variants could locate the absolute global optimum in all trials for this function. So in terms of error value there is no difference among these four algorithms. However, as the results in Table III reveal, in terms of required fitness evaluations there is significant difference. The newly proposed aDE algorithm required fewest fitness evaluations

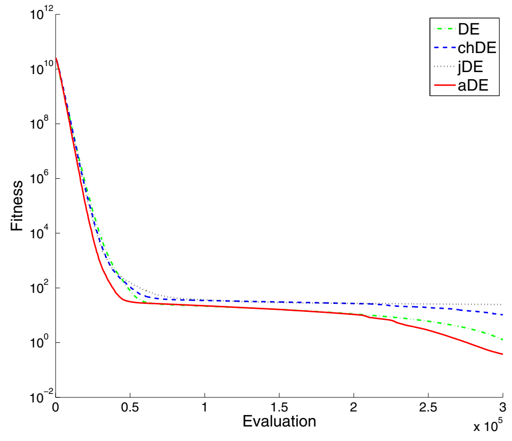

Fig. 7. Search traces for f_{ros} function

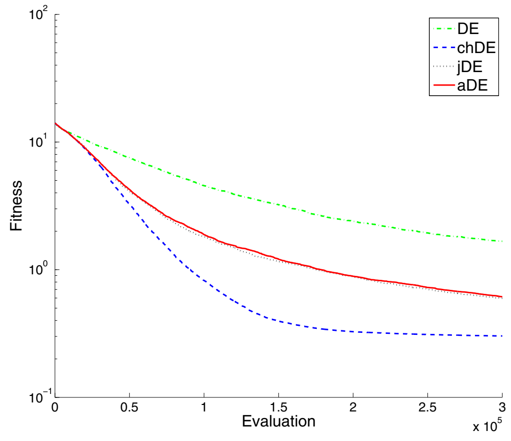

Fig. 9. Search traces for f_{scf} function

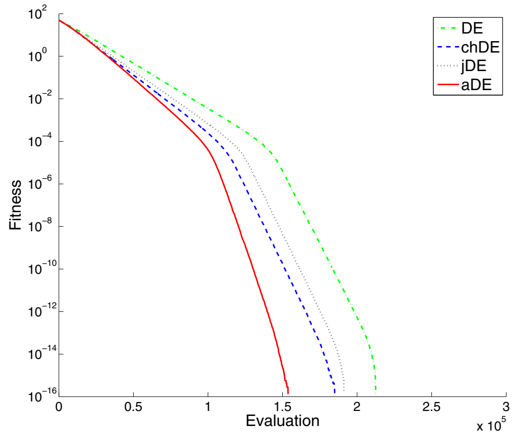

Fig. 8. Search traces for f_{wrs} function

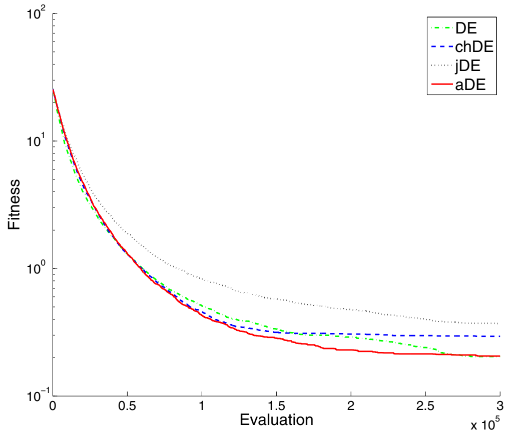

Fig. 10. Search traces for f_{sal} function

compared to DE, jDE and chDE, in locating the pseudo global optimum of this function. Although all the adaptive versions of DE showed improved performance relative to canonical DE for this function (see Fig. 8), the best performance was exhibited by aDE. Therefore, it may be stated that the parameter adaptation scheme in aDE was more effective for this function.

The Schaffer function (f_{scf}) is another non-separable problem with high multimodality. Consequently none of the DE variants could locate the global optimum of f_{scf}. However, in terms of error values, as shown in Table II, the best performance was displayed by chDE and the performance of jDE and aDE was very similar and better than DE. The convergence curves in Fig. 9 reveal that though chDE exhibited a steeper convergence curve at the beginning of the search, it started to converge quickly compared to aDE and jDE. Nevertheless, the final error value of chDE was significantly better than that of aDE and jDE in statistical measure (Table IV).

The last function in our benchmark suite is the Salomon function (f_{sal}) which is a non-separable, symmetric multimodal function. None of the studied algorithms was successful to locate the global optimum for this function. However, Table II shows that the best performance was exhibited by canonical DE, though not statistically significantly different from aDE (Table IV). The jDE and chDE algorithms showed a little deteriorated performance compared to DE. So it may be concluded that the proposed adaptation scheme is more robust compared to those in jDE and chDE.

In order to statistically validate the performance difference among the algorithms we analyzed the results using student's T-test at 99% confidence level. We calculated p-values both for error values and required fitness evaluations where appropriate and pair-wise compared aDE with other three algorithms. The results are summarized in Table IV where a $\Uparrow$ ($\Downarrow$) in column "ALG1 vs ALG2" (sub-column ErrorValue) indicates that ALG1 is significantly better than ALG2 (ALG2 is significantly better than ALG1) in terms of ErrorValue. A $\Leftrightarrow$ in the same column indicates that there is no significant difference between ALG1 and ALG2 and an empty cell indicates p-value was not calculable. Table IV shows that aDE is significantly better than DE in 9 benchmarks in terms of error value or/and fitness

TABLE IV
T-TEST ANALYSIS OF DE, jDE, chDE AND aDE PERFORMANCE

Fun	aDE vs DE		aDE vs jDE		aDE vs chDE	
	ErrorValue	Fitness Eval	ErrorValue	Fitness Eval	ErrorValue	Fitness Eval
f_{sph}	$\Uparrow$	$\Uparrow$	$\Uparrow$	$\Uparrow$	$\Uparrow$	$\Uparrow$
f_{ell}	$\Uparrow$	$\Uparrow$	$\Uparrow$	$\Uparrow$	$\Uparrow$	$\Uparrow$
f_{sch}	$\Uparrow$		$\Uparrow$		$\Uparrow$	
f_{ack}	$\Uparrow$	$\Uparrow$	$\Leftrightarrow$	$\Uparrow$	$\Leftrightarrow$	$\Uparrow$
f_{ras}		$\Uparrow$		$\Uparrow$		$\Downarrow$
f_{grw}		$\Uparrow$		$\Uparrow$		$\Uparrow$
f_{ros}	$\Uparrow$		$\Uparrow$		$\Uparrow$	
f_{wrs}		$\Uparrow$		$\Uparrow$		$\Uparrow$
f_{scf}	$\Uparrow$		$\Leftrightarrow$		$\Downarrow$	
f_{sal}	$\Leftrightarrow$		$\Uparrow$		$\Uparrow$	

evaluation required for global optimum. And aDE is better than jDE in 9 benchmarks in terms of error value or/and fitness evaluation. When compared aDE and chDE, aDE was found better than chDE in 8 benchmarks and chDE was better than aDE in the other two functions. Experimentation at $N = 100$ dimension resulted in more or less similar observation (results not presented here). So based on these empirical studies, it can be concluded that the newly proposed aDE is a more reliable and robust algorithm for global optimization compared to canonical DE, jDE and chDE.

VI. CONCLUSION

This work proposes a new adaptation mechanism for F and CR parameters of differential evolution (DE) algorithm. The notion of the presented adaptation scheme differs from the existed ones in that it preserves the effective parameter values and adjusts them only when they are not productive. Implementing the adaptation scheme at individual level a new variant of DE called aDE was presented.

In this work we methodologically analyzed the performance of the proposed aDE algorithm comparing with canonical DE and two other most widely used adaptive variants of DE, namely jDE and chDE (chaotic DE). We used most commonly used and recommended setting for canonical DE parameters and all these algorithms were implemented using the DE/rand/1/exp scheme of DE.

The empirical study with a standard test suite shows that aDE is a more robust and efficient algorithm compared to DE, jDE and chDE. The performance difference was also found to be statistically significant in almost all cases. This ascertain the effectiveness of the newly proposed adaptation scheme compared to the existing ones.

An in-depth study of the characteristics of the adaptation scheme and further improvement of the adjustment mechanism for the poor-performing parameters will be done in future.

REFERENCES

[1] R. Storn and K. V. Price, "Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces," ICSI, Technical Report TR-95-012, 1995.

[2] ——, "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.

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

[4] A. Qing, *Differential Evolution: Fundamentals and Applications in Electrical Engineering*. John Wiley & Sons, Singapore, 2009.

[5] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, vol. 9, no. 6, pp. 448–642, 2005.

[6] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A parameter study for differential evolution," in *WSEAS Int. Conf. on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, 2002, pp. 293–298.

[7] J. Liu and J. Lampinen, "On setting the control parameter of the differential evolution method," in *In: Proc. 8th Int. Conf. Soft Computing*, 2002, pp. 11–18.

[8] D. Zaharie, "Critical values for the control parameters of differential evolution algorithms," in *Proceedings of MENDEL 2002, 8th International Conference on Soft Computing*, 2002, pp. 62–67.

[9] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *IEEE Congress on Evolutionary Computation*, 2005, pp. 1785–1791.

[10] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "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.

[11] S. Das, A. Konar, and U. K. Chakraborty, "Two improved differential evolution schemes for faster global search," in *Genetic and Evolutionary Computation Conference (GECCO) Proceedings*, June 2005, pp. 991–998.

[12] Z. Guo, C. Bo, Y. Min, and B. Cao, "Self-adaptive chaos differential evolution," in *In Proceedings of International Conference on Natural Computation (ICNC)*, 2006, pp. 972–975.

[13] G. Yu, X. Wang, and P. Li, "Application of chaotic theory in differential evolution algorithms," in *In Proceedings of International Conference on Natural Computation (ICNC)*, 2010, pp. 3816–3820.

[14] L. D. S. Coelho, N. Nedjah, and L. de Macedo Mourelle, *Mobile Robots: The Evolutionary Approach*. Springer, 2007, ch. Differential Evolution Approach Using Chaotic Sequences Applied to Planning of Mobile Robot in a Static Environment with Obstacles, pp. 3–22.

[15] G. Wang and S. He, "A quantitative study on detection and estimation of weak signals by using chaotic duffing oscillators," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 50, no. 7, pp. 945–953, 2003.

[16] L. D. S. Coelho and V. C. Mariani, "Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 989–996, 2006.

[17] D. He, G. Dong, F. Wang, and Z. Mao, "Optimization of dynamic economic dispatch with valve-point effect using chaotic sequence based differential evolution algorithms," *Energy Conversion and Management*, vol. 52, no. 2, pp. 1026–1032, 2011.

[18] Y. Lu, J. Zhou, H. Qin, Y. Wang, and Y. Zhang, "Chaotic differential evolution methods for dynamic economic dispatch with valve-point effects," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 2, pp. 378–387, 2011.