A Survey on Adaptation Strategies for Mutation and Crossover Rates of Differential Evolution Algorithm

Dhanya M Dhanalakshmy#1, Pranav P#2, Jeyakumar G#3

# Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, Amrita University, India
E-mail: ‘md_dhanya@cb.amrita.edu; ’pranavprakash20@gmail.com; ‘g_jeyakumar@cb.amrita.edu

Abstract— Differential Evolution (DE), the well-known optimization algorithm, is a tool under the roof of Evolutionary Algorithms (EAs) for solving non-linear and non-differential optimization problems. DE has many qualities in its hand, which are attributing to its popularity. DE also known for its simplicity in solving the given problem with few control parameters: the population size (NP), the mutation rate (F) and the crossover rate (C). To avoid the difficulty involved in setting of suitable values for NP, F and C, many parameter adaptation strategies are proposed in the literature. This paper is to present the working principle of the parameter adaptation strategies of F and C. The adaptation strategies are categorized based on the logic used by the authors, and clear insights about all the categories are presented.

Keywords— Differential Evolution; Parameter Adaptation; Mutation Rate; Crossover Rate.

I. INTRODUCTION

Differential Evolution (DE) (proposed by Storn and Price [1],[2], a population based stochastic search method, is a very powerful algorithm in the repository of Evolutionary Algorithms (EAs). The performance efficacy of DE, comparing with other EAs, for solving real time and benchmarking problems which are non-linear, complex and high dimensional over continuous domain has been well proved in its literature [3]. The algorithmic structure of DE is similar to other EAs. However, unlike other EAs, DE uses very few control parameters: the population size (NP), the mutation rate (F) and the crossover rate (C). The efficiency and accuracy of DE algorithm is more sensitive to the values chosen for these few parameters.

The successful convergence of DE to the global optimum solution, in its evolutionary search for solving the given problem, is largely depend on suitable selection of values for these control parameters. Finding the suitable values for these control parameters, before starting the search, is a difficult task as it will differ from problem to problem. A poor choice of these values will result in the poor accuracy of the algorithm which is not acceptable. There is no single perfect method or standard available for selecting values for these control parameters. Hence, the process of tuning these control parameters along with the search became an attractive area of research for the researchers’ community working in DE. This results numerous adaptation strategies proposed for NP, F and C in DE literature. Subsequently, this has become a challenge to the practitioners, researchers and users of DE to choose right adaptation strategies for each of the control parameter to solve the problem at their hand. Resolving this challenge is taken as the aim of this paper.

The objective of this paper is to provide the readers with brief insight about various adaptation strategies proposed by researchers for adapting F and C. It is obvious that the number of researchers working in DE, particularly in the parameter adaptation of DE, is increasing day after the other. This paper is intended to provide them with summary of various adaptation strategies exist in DE literature for tuning F and C.

II. DIFFERENTIAL EVOLUTION ALGORITHM

For a search method to be efficient and reliable, it has to cover the entire search space. Differential Evolution starts its search of global solution for the given optimization problem, with randomly selected NPD-dimensional population vectors (individuals/candidates). The initial population is chosen in such a way that the individuals are initialized randomly in order to cover the entire search space. The population vector is represented as \( X_{iG} = \{x_{i,1G}, x_{i,2G}, \ldots, x_{i,DG}\} \), \( i \) ranges from 1 to NP, \( G \) represents the generation and \( D \) represents the number of parameters for each individual (ie, dimension of the problem).
The three evolutionary processes involved in DE are mutation, crossover and selection. Among these the mutation and crossover are called variation operators, which brings changes in the population by altering the values of the components of the individuals in the population. The changes made by these operators create new candidates in the population, thus increasing the diversity of the population. Hence they attributed to exploration phase of the search. On the other hand, the selection process selects the best candidate from a set of candidates. Thus it is for the exploitation phase of the search.

At first the mutation process takes place. From the initial population the mutation process generates a mutated population. This process is termed as *Differential Mutation* in DE. The mutation process chooses three random candidates (say $C_1$, $C_2$ and $C_3$) from the population, and generates a mutant vector ($MV$) as

$$MV = C_k + F \times (C_2 - C_3)$$  \hspace{1cm} (1)

where $F$- mutation rate or scaling factor. In Eqn (1), the scaled difference of $C_2$ and $C_1$ is added to $C_i$ (also known as base vector). There exist many ways to choose the base vector and the other pair of vectors for mutation. Based on that, there are many mutation strategies available for DE. The critical parameter in the mutation process is the scaling factor $F$. One mutant vector is generated for each vector population (also known as target vector ($TaV$)) in the current, which results mutated population with NP mutant vectors.

Secondly, the *crossover* process generates the trial vector ($TrV$) population. This process recombines each of the $TaV$ in the current generation with its corresponding $MV$ to produce the $TrV$. The values from the parameters of $TaV$ and $MV$ are used to generate a $TrV$. The crossover process results one $TrV$ for each $TaV$. The *crossover* process determines how much information the trial vector (child) inherits from its parents (target and mutant vectors).This is determined by the control parameter called the crossover rate ($C_i$). The most common two crossover strategies of DE are *binomial crossover* and *exponential crossover*. The equation for binomial crossover is given in equation (2). The crossover process also repeated for all the pair of target and the corresponding mutant vector, which results a population of NP trial vectors.

$$TrV = \begin{cases} MV & \text{if } C_i \geq \text{Rand}(0,1) \\ TaV & \text{Otherwise} \end{cases}$$  \hspace{1cm} (2)

Next, a selection process is carried out between each target vector in the current population and their corresponding trial vectors. DE uses one-to-one tournament selection based on the fitness values of the candidates (vectors). The better candidate out of the two will have the privilege to move to the next generation.

Each generation of DE’s search process include these three evolutionary processes (Mutation, Crossover and Selection). The whole process is repeated for $G$ (maximum number of generations) number of generations, which is considered as one run in DE experiment. The best solution obtained at the end of the run is the solution obtained by DE for the given problem, at that particular run. Since all the stages of evolutionary process in DE (in fact any EA) involves randomness, the average performance of DE algorithm in its many runs is used for reporting its performance.

### III. CONTROL PARAMETERS OF DE

Understanding the influence of the parameters of DE (*mutation rate* ($F$), *crossover rate* ($C_i$) and *population size* ($NP$)) is essential to know the adaptation strategies available for them. This section presents the role of $F$, $C_i$ and $NP$.

The mutation process is to alter the values of the components of each of the candidate in the population. The mutation of DE is called as *differential mutation*, since it uses weighted differences of candidates to perform mutation for the current candidate. One among the unique feature of DE is its differential mutation. The mutation process can be understood from the Figure 1.

![Fig. 1. The Differential Mutation of DE.](image)

The Mutation rate ($F$) (also known as scaling factor, amplification factor, mutation step size or mutation constant) is to scale the distance between the pair of vectors $C_2$ and $C_3$. This scaled difference is added to the base vector $C_i$. Thus, the mutation rate is used to control the amplification of the difference vector. Hence, a small value of $F$ will lead to premature convergence whereas a larger value will result in a slower convergence. It controls the range of space where the mutant vectors are generated. Thus, it plays an important role in changing diversity in the population.

In classical DE algorithm the value for $F$ is taken as any real value in the range of 0 to 1. Keeping the value of $F$ as constant will deteriorate the diversity of the population during the search, because all the vectors will be created by same difference vector components. So in order to avoid this, many classical DE implementations follow a different strategy where $F$ will be considered as a random number within the range of [0.2, 0.8]. This ensures that the diversity loss during the search is avoided.

In natural evolution, the crossover process is to create children by inheriting genetic properties from parents. It holds good for DE search also. In this process of genetic inheritance, to get diversified candidate from the parent, the parameter crossover rate ($C_i$) is used. As similar to $F$, $C_i$ also a real valued parameter in the range of 0 to 1. It is used to identify the parameters to be inherited from the parents. Crossover rate ($C_i$) controls the number of elements that the trial vector will inherit from mutant vector and target vector. Thus it defines how different the child vector is from the
parent vectors. In other words, it ensures diversity in the newly created population. Finding the right value of this control parameter is a difficult task as a slight change in the $C_r$ value will affect the efficiency of the algorithm. When $C_r$ value is approximately equals to 0, $DE$ makes small explorative moves with higher probability of making improvement. When $C_r \approx 0.9$, $DE$ makes large explorative moves which helps to perform a more fine-grained search in the solution space and yield large improvements in solution quality.

The third control parameter $NP$ also has significant impact on performance of $DE$. If $NP$ is small, the search may end in premature convergence, and if it is large the search will take long time to converge. Hence a moderate value for $NP$, to avoid the premature convergence and stagnation, is acceptable for successful $DE$ search.

There are many methods proposed for adapting $F$ and $C_r$, many of them were found to be performing better when compared with classical $DE$. However, less works are reported in the literature for adapting $NP$. Hence this paper considers, hereafter, to discuss the existing adaptation strategies for $F$ and $C_r$.

IV. ADAPTATION STRATEGIES FOR $F$ AND $C_r$

To plan for a suitable adaptation strategy of the control parameters, it is necessary to understand the influence of each of the parameters in the performance of the algorithm.

There are many works reported in literature to discuss the influences of the control parameters. It was in the year 2001, Zaharie [4] was one among the few who started analyzing the possible effect of these control parameter values on $DE$ and their critical values. The approach was both a theoretical and empirical study on how the control parameter values are related with population variance of $DE$. An equation to measure the critical values of control parameters was derived. The equation derived by Zaharie was $2Fp-2p/m + p/m+1=0$, where $F$ is the scaling factor, $m$ is the population size, and $p$ is the crossover rate. The value ($F$ and $P$) that satisfies this equation was considered critical values. Tremendous efforts have been put by the researcher to analyze the role of each of the $DE$ control parameters. Presenting about them is not in the scope of this paper.

Since the impact of the control parameters $MutationRate$ ($F$) and $CrossoverRate$ ($C_r$) on the performance of the algorithm is very high, many control parameter adaptation techniques has been put forwarded over the years. All of them have been proved as effective and improving the performance of Differential Evolution algorithm in both converge speed and solution accuracy, compared to the classical $DE$.

The objective of this chapter is to present a brief insight about various adaptation strategies exist in literature. To increase the readability of the paper, the existing adaptation strategies of $F$ and $C_r$ are categorized in to four groups with respect to the algorithmic methodology followed by the authors. The details of the categories [57] are

- **Category 1:** Classical Approach

The strategies in this category use mathematical equations to update the values of the control parameters. This updation is done for every generation or at required time.

- **Category 2:** Encoding of control parameters

Another way of adapting control parameter is to encode the control parameters along with the parametric values of the candidates of the population. Hence, the control parameters also evolve as similar to other parameters. The adaptation strategies following this methodology are grouped under this category.

- **Category 3:** Deriving from History or Pool

The strategies which use previous information about the performance of the algorithm in the evolutionary search are grouped under this category. The algorithms which maintain pool of values for the control parameters also grouped in this category.

- **Category 4:** With added logic

The strategies which use some additional technique or algorithm to adapt the parameters are discussed under this category.

A. Classical Approaches

With the understanding of role of $F$ and $C_r$ in the mutation and crossover processes, the works considered in this category uses mathematical equations derived by the authors to update the values of $F$ and $C_r$.

In $SaDE$ [5], proposed by A.K Qin and P.N.Suganthan, an adaptive logic for mutation strategy is presented. The mutation strategy is decided based on the success rate calculated for them in the learning period. The $F$ and $C_r$ values also calculated differently for each of the mutation strategy. For every individual $i$ in the population the $F$ and $C_r$ values for the chosen mutation strategy $k$ is calculated as follows

$$F = \text{rand}(0.5,0.3)$$

$$C_r = \text{uniform\_rand}(C_r, 0.1)$$

where $C_r$ is calculated from a success rule as the mean of $C_r$. This is followed by many researchers for different applications of $DE$ [6].

The $JADE$ [7][8][9][10]) proposed by Jingqiao Zhang and Arthur C. Sanderson in the year 2007, introduced a new mutation strategy based on the information obtained about the search progress direction. The values of $F$ and $C_r$ are generated newly for each generation with Cauchy and normal distribution. With the initial values of 0.5, the new values at generations are computed as follows.

$$F = (1-c) \cdot F_r + c \cdot \text{mean}(F)$$

$$C_r = (1-c) \cdot C_r + c \cdot \text{mean}(C_r)$$

where $c \in (0,1)$, is a constant.

$S_F$ and $S_{C_r}$ represent the mean of successful values for crossover rate and scaling factor, respectively. Many other modified $DE$ were introduced which borrowed this concept [10]. In the year 2008, Wu Zhi-Feng proposed a new version of $DE$ called $AdaptDE$ [11]. The fitness values of the trial vector and the target vector are used to find the values for $F$ and $C_r$. Also, the control parameter values for the $G+1^{th}$ generation is calculated by using their values in $G^{th}$
where, $C_{rmax}$ and $C_{rmin}$ are the maximum and minimum crossover rate of generation $G$. The $r_1, r_2$ are two fixed values (Similar to jDE) and $rand_0, rand_2$ and $rand_4$ are uniform random numbers in the range (0,1).

In the same year, Ali W. Mohammed et al. proposed ADE [16]. This paper introduces an alternative differential evolution (ADE) algorithm. In ADE a new mutation scheme is proposed and control parameters are adapted using a defined equation for both $F$ and $C_r$. Two values of $F$ are defined, one for the local mutation scheme and the other one for global mutation scheme. Keeping in mind the fact that $C_r$ must start with a small value and must extend to a larger value as the generations increases, the authors framed an equation to $C_r$ as follows,

$$C_r = C_{r_{min}} + (C_{r_{max}} - C_{r_{min}}) \times (1 - \frac{G}{G_{max}})^{1/2}$$

where $G$ is the current generation and $G_{max}$ is the maximum number of generations.

The authors also mentioned the optimum values for $C_{rmin}$, $C_{rmax}$ and $K$ as 0.1, 0.8, and 0.4 respectively.

Ali W. Mohammed et al. also proposed EDE (Effective Differential Evolution) [17]. A simple method for choosing the values for the control parameters is used. The values for $C_r$ and $F$ are chosen empirically from the range of [0.5, 0.9] and [0.2, 0.8], respectively. This range ensured that there will be adequate exploitation and exploration during the search.

In 2013 TLBSaDE was introduced by Subhodip Biswas et al. [18]. TLBSaDE borrows the basic concept from SaDE. It is based on concept of how learners gain knowledge in class from a teacher. The strategy for adapting control parameters is used as follows: The value of $F$ is taken from a normal distribution with mean and standard deviation of 0.5 and 0.3, respectively. The value of $C_r$, similar to SaDE, is taken from

$$F_{k,G+1} = \text{uniform}_\text{rand}(0.1, 1.0)$$

$$C_{r_{(G+1)}} = \text{uniform}_\text{rand}(0.0, 1.0)$$

In 2012, PengGuo et al. proposed SelfDE-F [15]. In SelfDE-F, a secondary population is created with the individuals that were discarded during the selection process of the DE algorithm. The adaptive method for finding control parameter values for each generation was framed as follows

$$F_{i,G+1} = \begin{cases} 
  w_M \times F_{\text{best}_{i,G}} + (1 - w_M) \times \text{rand} & \text{if } C_1 \\
  F_{jDE} & \text{otherwise}
\end{cases}$$

$$C_{r_{(G+1)}} = \begin{cases} 
  C_{\text{rand}} + \text{rand} \times C_{r_{\text{rand}}} & \text{if } C_2 \\
  C_{\text{rand}} & \text{otherwise}
\end{cases}$$

where $w_M = e^{\frac{-\text{fitness}(G+1) - \text{fitness}(G)}{\text{fitness}(G+1) - \text{fitness}(G+1)}}$

$F_{\text{best}_{i,G}} = \text{Scaling factor of the candidate with best fitness value.}$

$CR_{max,G}$ and $CR_{min,G}$ are the maximum and minimum crossover rate of generation $G$. The $r_1, r_2$ are two fixed values (Similar to jDE) and $rand_0, rand_2$ and $rand_4$ are uniform random numbers in the range (0,1).

It is changed adaptively over the generations.

The mean fitness value of a generation is calculated. To find the $F$ and $C_r$ values for $G+1^{th}$ generation, the mean fitness value of previous generation ($G^0$) is compared with the mean fitness value of $G+1^{th}$ generation, and

If $\text{Meanfitness}_{G^0} > \text{Meanfitness}_{G+1}$ then

$$F_{G+1} = F_G$$

$$C_{r_{G+1}} = C_{r_G}$$

Else

$$F_{G+1} = F_G + \text{rand} \times (0.1) \times (C_{\text{rand}} - F_G)$$

$$C_{r_{G+1}} = C_{r_G} + \text{rand} \times (0.1) \times (C_{\text{rand}} - C_{r_G})$$

NasimulNoman, in [14], introduced an approach to use the fitness value of the child and the average fitness value of the population to update the control parameters values. It is done as follows, If fitness values of child is less than average fitness value then the control parameter values $F$ and $C_r$ at $G^0$ is taken for their $(G+1)^{th}$ generation. Otherwise,

$$F_{i,G+1} = \text{uniform}_\text{rand}(0.1,1.0)$$

$$C_{r_{(G+1)}} = \text{uniform}_\text{rand}(0.0, 1.0)$$

And Crossover rate is found using the rule,

$$C_{r_{G+1}} = \begin{cases} 
  C_{r_G} + \text{rand} \times (0.1) & \text{if } F < \text{rand}1 \\
  C_{r_G} & \text{otherwise}
\end{cases}$$

where, $rand_0, rand_2, rand_4$ are random numbers within the range [0,1] and $r_1$ and $r_2$ are probabilities for adjusting $F$ and $C_r$. The authors preferred a value of 0.1 for both.

In 2009, RadhaThangaraj et al. introduced ACDE [12]. In ACDE, the whole adaptation process is based on few simple rules. Scaling factor for an individual $i$ is defined as

$$F_{G+1} = \begin{cases} 
  F_G + \text{rand} \times \frac{max(C_{r_{G+1}} - min(C_{r_{G+1}}))}{1} & \text{if } F < \text{rand}2 \\
  F_G & \text{otherwise}
\end{cases}$$

And Crossover rate is found using the rule,

$$C_{r_{G+1}} = \begin{cases} 
  C_{r_G} + \text{rand} & \text{if } F < \text{rand}4 \\
  C_{r_G} & \text{otherwise}
\end{cases}$$

Initially, the values of $F$ and $C_r$ will be fixed as 0.6 and 0.1, respectively.

It is changed adaptively over the generations.

The mean fitness value of a generation is calculated. To find the $F$ and $C_r$ values for $G+1^{th}$ generation, the mean fitness value of previous generation ($G^0$) is compared with the mean fitness value of $G+1^{th}$ generation, and

If $\text{Meanfitness}_{G^0} > \text{Meanfitness}_{G+1}$ then

$$F_{G+1} = F_G$$

$$C_{r_{G+1}} = C_{r_G}$$

Else

$$F_{G+1} = F_G + \text{rand} \times (0.1) \times (C_{\text{rand}} - F_G)$$

$$C_{r_{G+1}} = C_{r_G} + \text{rand} \times (0.1) \times (C_{\text{rand}} - C_{r_G})$$

NasimulNoman, in [14], introduced an approach to use the fitness value of the child and the average fitness value of the population to update the control parameters values. It is done as follows, If fitness values of child is less than average fitness value then the control parameter values $F$ and $C_r$ at $G^0$ is taken for their $(G+1)^{th}$ generation. Otherwise,
a normal distribution $N(C_{\text{m}},0.1)$. The value for $C_{\text{m}}$ (mean) is kept as 0.5 and the standard deviation is 0.1.

SAMDE [19] was also introduced in the year 2013 by Xu Wang et al. The SAMDE is similar to JADE. Scaling factor is found using Cauchy’s Distribution with mean $\mu_F$, 

$$F = \text{rand}() \mu_F \text{ } (22)$$

$\mu_F$ is initialized to 0.5 and then it is changed as, 

$$\mu_F = (1 - c) \mu_F + c \text{ } \text{mean}(S_F) \text{ } (23)$$

where $\text{mean}(S_F)$ is the Lehmer mean.

Crossover rate is found from a normal distribution of mean $\mu_{CR}$ and standard deviation 0.1.

$$CR = \text{rand}() \mu_{CR} \text{ } (24)$$

$$\mu_{CR} = (1 - c) \mu_{CR} + c \text{ } \text{mean}(S_{CR}) \text{ } (25)$$

where $\text{mean}(S_{CR})$ is the arithmetic mean.

Rammohan Mallipeddi and Minho Lee in the same year proposed ESMDE [20], an evolving surrogate (substitute) model-based DE. In ESMDE, based on the current population a surrogate model is created and this is used for selecting appropriate parameter setting so as to creating better off springs during further stages of evolution. Similar to other approaches, the mutation strategies are selected according the concept of pooled values. Here $F$ (scaling factor) is selected randomly within the range of [0.5, 1.0] which ensures that there will be adequate exploration alongside with exploitation. Similarly, $C_r$ value is also generated randomly from a range of [0,1].

Quizhen Lin et al. proposed an adaptive algorithm [21], in which $C_r$ is found using a framed equation which ensures that the $C_r$ value will be very large (approximately equal to 0.9) initially and as the generations increases the value decreases and will be stagnant in the range of 0.1 to 0.2. This ensures that the explorative steps will be very large during the initial phase of the algorithm to favour the global search. As the generations goes the explorative steps will be reduced and the search will be done near local. This ensured good performance of the algorithm. The ScalingFactor is found using a Cauchy's Distributions within the range of (0.5, 0.1). The value for $F$ is selected from a set of successful values collected for $F$.

Miguel Leon et.al proposed a greedy adaptation of control parameters of DE [22]. Here a greedy search will be performed in learning periods that are successive so as to favour continuous and dynamic adjustments of the control parameters $F$ and $C_r$. The whole procedure is as follows, initially the $F$ value is set to 0.5 and two neighbours are defined in such a way that a difference $d_x$ is added and subtracted to the initial $F$, i.e., $(F - d_x \text{ and } F + d_x)$. The initial Crossover rate is set to a Cauchy’s Distribution with its centre at 0.5 ($\mu_{CR}$) and scale of 0.2. Two neighbours of $C_r$ are $C_{r+} - d_x$ and $C_{r+} + d_x$. During the predetermined learning period, every candidate and its neighbouring candidate will have a probability of 1/3 in order to get sufficient number of usages. At the end of the learning period, the best will replace the worst.

Qinqin Fan and Xuefeng Yan proposed another adaptation method [23] to their self-adaptive DE strategy (named as SDE). This is an algorithm with zoning evolution of control parameters and adaptive mutation strategies, called as ZEPDE. They have defined their own methods to identify the best possible mutation strategies for the DE algorithm. The parameter adaptation is done as follows. The total region of $F$ and $C_r$ are divided into similar sized four areas. Then at each region the count of control parameter combinations are noted. If the offspring in each zone has a better fitness function value, then it is taken into account that the control parameter combination which gave the best fitness value is the elite one or can be called as Elite Control Parameter Combination (EPC). Assume at the $h^{th}$, the weighted value of each $EPC$ is computed as follows,

$$W^h_{EPC} = \frac{f(X_i^h) - f(X_{g,best}^h)}{f(X_{g,best}^h) - f(X_{g,2nd,best}^h)} \text{ } (26)$$

where $N^h_{EPC} = 1,2,...N^h_{EPC, elite}$

Now the weighted average of control parameters in the $h^{th}$ region is calculated as

$$F^h_{EPC} = \sum_{N^h_{EPC}} W^h_{EPC} \text{ } (27)$$

and 

$$C_{r,h} = \sum_{N^h_{EPC}} W^h_{EPC} \text{ } (28)$$

Once the weighted average is calculated the control parameters for the $h^{th}$ generation is calculated using a cauchy’s distribution with mean $F_{EPC}^h$ and $C_{r,h}$ and standard deviation $(0.55-0.3 *(1-G/Gmax))$.

Xiaowei Zhang and Sanyang Liu proposed APFDE [24] in 2011, drawing inspiration from the theory of electromagnetic. They calculated the charge $Qi$ for candidate $Xi$ based on its objective function value and the objective function value of the best candidate in the current generation using the equation given below

$$Q_i = e^{-\frac{(f(X_i) - f(X_{g,best})}{D}} \text{ } (29)$$

Later, the equation was modified as shown below:-

$$Q_i = \frac{f(X_i) - f(X_{g,best})}{f(X_{g,2nd,best}) - f(X_{g,best})} \text{ } (30)$$

where $D$ is the problem dimension and $NP$ is the population size. The Mutant Vector generation in normal DE can be written as follows

$$V_i = X_i + F(X_i - X_{g,best}) \text{ } (31)$$

Using the above derivations, $F$ is replaced by $Q_{12}$ and $Q_{13}$ respectively. Taguchichi method along with 2-level orthogonal Array is used to set the value of $C_r$. 

Islam et al. [25] proposed MDE_pBX that introduced current-to-g_best mutation scheme and p-best crossover scheme, along with schemes for updating $F$ and $C_r$ in each
generation. The Scale Factor for the candidate \( F_i \) is randomly picked from a Cauchy distribution with location Parameter \( F_m \) and scale parameter 0.1. The list of all \( F \) values that generated better trial vectors are stored in a set \( F_{\text{success}} \). This is updated using the following equation

\[
F_{\text{success}} = F_{\text{success}} \ast W_F + \left( \sum_{j=1}^{g} \text{random} \left( 1- \frac{\Delta F}{\text{CR} \times F_{\text{success}}} \right) \right)^{\frac{1}{2}} \times (1 - W_F) \tag{32}
\]

Where \( W_F \) is \( 0.8 + 0.2 \times \text{random} \left( 0,1 \right) \). \( C_i \) is selected randomly from a Gaussian distribution with mean \( C_m \) and Standard deviation 0.1. The \( C_m \) value is updated in each generation in a similar way as that of \( F \) according to the equation given below

\[
C_{\text{success}} = C_{\text{success}} \ast W_C + \left( \sum_{j=1}^{g} \text{random} \left( 1- \frac{\Delta C}{\text{CR} \times C_{\text{success}}} \right) \right)^{\frac{1}{2}} \times (1 - W_C) \tag{33}
\]

where \( W_C \) is \( 0.9 + 0.1 \times \text{random} \left( 0,1 \right) \). \( C_{\text{success}} \) is the set of all successful \( C \) values.

Yet another scheme for adapting \( F \) and \( C \) was proposed in [26], in which the candidates are put into two sorted lists - the first one in descending order based on objective functions and the second one in ascending order based on each candidate’s distance from the best candidate. Then the sum of absolute differences between these two ranks for each candidate is calculated, which is called as Indicator of Optimization State (IOS). Also \( IOS_{\text{min}} \) is set to 0 and \( IOS_{\text{max}} \) is calculated using the following equation

\[
IOS_{\text{max}} = \left\{ \begin{array}{ll}
\frac{\text{NF}^2}{2} & \text{if } \text{even} \text{ \ NF is even} \\
\frac{\text{NF}^2}{2} & \text{if } \text{odd} \text{ \ NF is odd}
\end{array} \right.
\tag{34}
\]

Then IOS is normalized and this normalized value is used to decide whether to explore or exploit. If exploration is to be done, \( F_{p-1} \) is increased by \( 1/10^6 \) of \( \Delta F \) and \( CR_{p-1} \) is decreased by \( 1/10^6 \Delta CR \) respectively; otherwise \( F_{p-1} \) is decreased by \( 1/10^6 \) of \( \Delta F \) and \( CR_{p-1} \) is increased by \( 1/10^6 \Delta CR \) respectively. \( \Delta F \) and \( \Delta CR \) are computed based on IOS, \( IOS_{\text{max}} \) and \( IOS_{\text{min}} \) based on the equation

\[
\Delta F, \Delta CR = \left\{ \begin{array}{ll}
\text{IOS}_{\text{min}} & \text{for exploration step} \\
\text{IOS}_{\text{max}} & \text{for exploitation step}
\end{array} \right.
\tag{35}
\]

The mathematical equations used in the studies reported in this section are derived / defined by the authors, based on their understanding about the control parameters and their influences in \( DE \) algorithm. It is worth noting that such equations also adds few new terms to it, which are again to be studied further to set proper value for them. This indirectly increases the complexity of parameter adaptations. To avoid this there exist many parameter adaptation strategies for \( DE \) control parameters in literature. They allow the values of control parameters also to evolve, as similar to parameters of the candidates, to be better for next generation. Those adaptation strategies encode the control parameters in the parametric representation of the candidates in the population. The next section discusses the strategies with such encoding scheme.

B. Control Parameter Encoding

To avoid inclusion of additional parameters in adapting the required parameters, the idea of encoding the control parameters with the individual candidates in the population arose. This encoding let the control parameters also to evolve along with other parts of the candidates. At the required stages the \( F \) and \( C \) values are selected suitably from the evolved values of them. Hence, these strategies include efficient selection mechanism to consider the evolved values of \( F \) and \( C \) available at the parent candidates of mutation and crossover. The works similar to this strategy are discussed in this chapter.

Mahamed G. H. Omran et al. introduced a self-adaptiveDE [27] in 2005. This algorithm uses the differential mutation mechanism to find the new values of \( F \) and \( C \), each individual \( i \) in the population is encoded with \( F_i \) and \( C_i \), and these values are calculated as follows

\[
F_i = F_{i,\text{old}} + \text{random} \left( 0,1 \right) \left( F_{\text{best}} - F_{i,\text{old}} \right) \tag{36}
\]

where \( i_1 \) and \( i_2 \) are random and distinct candidates chosen with a uniform distribution \( U(1,..,NP) \).

Another self adaptive \( DE \) named \( SADE\_ALM \) (Self Adaptive Differential Evolution with Augmented Lagrange Multiplier) [28][29] was proposed by C. Thitithamrongchai and B. Eua-Arporn in 2006. In the \( SADE\_ALM \) the \( F \) and \( C \) are encoded in the first two positions of the candidates. The \( F \) and \( C \) values are initialized as follows

\[
F_i = F_{i,\text{old}} + \text{random} \left( 0,1 \right) \left( F_{\text{best}} - F_{i,\text{old}} \right) \tag{37}
\]

where \( i_1 \) and \( i_2 \) are non-equal indices between 1 and \( NP \). The crossover process is done as follows:

\[
F_i = \left\{ \begin{array}{ll}
F_{i,\text{old}} & \text{if } \text{random} < CR_{\text{old}} \text{ or random} < CR_1 \text{ or random} < CR_2 \\
F_{i,\text{new}} & \text{otherwise}
\end{array} \right.
\tag{38}
\]

Finally, at required points, the \( F \) and \( C \) values are chosen from the best candidate of the population. Amin Nobakhti and Hong Wang in 2006, proposed an adaptive \( DE \) [30][31] with control parameter adaptation, mainly for the mutation rate \( (F) \). Each population vector is assigned with their own value of \( F \), which is initialized by a uniform distribution bounded by \( F_{\text{min}} \) and \( F_{\text{max}} \). During the process of evolution, once the trial vector has been created and is found to be better than the target, the trial vector will inherit the value of \( F \) from the target. After every fixed ‘k’ generations, the entire population is analyzed for accumulated improvements and is sorted accordingly. From this sorted population, a threshold value is identified which represents the accumulated
improvements of at least half the population. For all the individuals, whose $R_i$ value is greater than this threshold, $F$ values are retained and for others new $R_i$ values are randomly generated.

In the year 2007, J. Brest et al. put forwarded the jDE algorithm [27],[32],[33],[34],[35]. In jDE also, the control parameters $F$ and $C_r$ are encoded with the population along with its genes. These values are adapted during every generation according to two other fixed values $\tau_1$ and $\tau_2$. The value of $F$ in the $(G+1)^{th}$ generation will be same as the value in $G^{th}$ generation if a randomly created number within the range (0,1) is greater than $\tau_1$, else $F$ value is found as

$$R_{i,g+1} = R_{i,g} + \text{rand} \times \delta_{i,g} \tag{43}$$

Suffix ‘t’ and ‘u’ stands for lower and upper values of $F$. Similarly, for $C$, if $\tau_u$ is less than randomly generated value then the previous generation value is retained, else

$$C_{R,i,g+1} = \text{rand}(0,1) \tag{44}$$

An enhanced version jDE, named jDE-2 [36], was introduced later with an added concept for keeping the bound-constraints problem feasible. But the parameter control part was kept same as that of jDE.

Brest also introduced a different variant of the above called jDEdynNP-F [37] in 2008, where along with control parameter adaptation, population size reduction mechanism is also implemented. The $F$ and $C_r$ adaptation of the proposed algorithm remain the same as that of its previous version. Similar adaptation mechanism for $F$ and $C_r$ is followed by Zhong-bo Hu et al. [38].

Also in the same year Chukiat Worasucheep, proposed wDE [39]. In wDE, a separate strategy adaptation and parameter adaptation is introduced. In parameter adaptation, each individual ‘i’ is extended with corresponding $C_r$ and $F_i$ which are uniformly initialised within the range [0.1] and [0.2], respectively, at the beginning. A fixed number of generation period (learning period) is considered and adaptation of these control parameters happen after these fixed number of generations. Also two variable $nsi$ and $nfi$ are introduced which indicates the number of success and failure for a particular individual $i$ for entering into the next generation with respect to the learning period. The probability to adapt $P_{i}$ for a particular individual $i$ and the average of pass probabilities ($PP_{avg}$) are calculated. Now, for all those individuals, whose probability of pass is below the probability pass average, the control parameters are updated as follows

$$R_i = \text{uniform} \text{rand}(0,2) \tag{45}$$
$$C_{R,i} = \text{uniform} \text{rand}(0,1) \tag{46}$$

AlesZamuda and BorkoBoskovic in 2007 came up with DEwSAcc [40], Differential Evolution With Self-adaptation and Cooperative Co-evolution. In DEwSAcc also, as similar to other works in this category, all the individuals in the population are extended to include their own control parameters, $F_i$ and $C_r$. The control parameter value of the next generation $(G+1)$ depends on the values of these control parameters of the current generation $(G)$. $F$ and $C_r$ are updated as follows

$$R_{i,g+1} = R_{i,g} + \text{rand} \times \delta_{i,g} \tag{47}$$
$$C_{R,i,g+1} = \text{adapt} \tag{48}$$

where, $\tau$ represents the learning period which is $1/\sqrt{D}$.

In 2008, Omar S. Soliman and Lam T. Bui came up with a self-adaptive strategy to use Cauchy distribution [41]. The control parameters for each individual are encoded along with them. For an individual $i$, scaling factor is found by

$$R_{i,g+1} = \begin{cases} C_{R,i,g} \times \text{rand} & \text{if } \text{rand} \leq \tau_1 \\ C_{R,i,g} \times \text{rand} & \text{otherwise} \end{cases} \tag{49}$$

where, $\delta_{i,g+1} = \delta_0 = \delta_0 \times \text{rand}$ and $\delta_0$ and $\delta_0$ are the lower and upper limit of possible values for scaling factor. Crossover rate is found as follows:

$$C_{r,g+1} = \begin{cases} \text{rand} & \text{if } \text{rand} \leq \tau_2 \\ C_{r,g} & \text{otherwise} \end{cases} \tag{50}$$

Grant Dick [42] in 2010 defined SaNSDE (self-adaptive neighbourhood search differential evolution) which uses the neighbourhood search which is one of the core concepts of Evolutionary Computing. Control Parameters are added to each individual. It works as follows, the first step is to initialise the $CR_m$ value to 0.5. Then similar to above it is found during each generation using a normal distribution with mean value $CR_m$ and standard deviation 0.1 for each of the individual. $F$ value is found as follows,

$$F_i = \begin{cases} \beta_{i,0} \times \text{rand} & \text{if } U_i(0.1) < f' \\ \beta_{i,0.1} \times \text{rand} & \text{otherwise} \end{cases} \tag{51}$$

where $U_i(0.1)$ is a uniform random number in range [0,1).

The SaFDE [43] proposed by Teng NgaSing et al. encodes scale factor inside each candidate. Initial population has $F$ randomly initialized for each candidate. During trial vector generation, if the random number generated is less than cross over probability, trial vector’s scale factor is also updated in the same way as the other genes using the differential mutation as given below

$$\text{TrialVector} \times F = X_1.F + \alpha_{DE} \times (X_2.F - X_3.F) \tag{52}$$

Here $\alpha_{DE}$ is a randomly chosen value between 0 and 1. If the calculated value of $F$ goes beyond the limits [0,1], then it is randomly re-initialized. Authors have self adapted only $F$ and $C_r$ and they are selected randomly from [0.1, 1.0].

In the year 2013, Ming Yang et al. proposed a variation of jDE called PA-jDE [44] that does a population adaptation along with adaptation of $F$ and $C_r$. $F$ and $C_r$ is encoded with
each individual. $F_i$ and $C_i$ are updated in each generation based on two thresholds $T_1$ and $T_2$ as follows:

$$F_{i_{1:1}} = \begin{cases} F_i + \text{rand} (r_{F_i} - F_i) & \text{if } r_n < 0.2 \\ C_{R_{i_{1:1}}} = C_{R_i} + \text{rand} (C_{R_i} - C_{R_0}) & \text{otherwise} \end{cases}$$

$$C_{R_{i_{1:1}}} = \begin{cases} C_{R_i} - \text{rand} (C_{R_i} - C_{R_0}) & \text{if } r_n < 0.2 \\ \text{rand} (C_{R_0} - C_{R_i}) & \text{otherwise} \end{cases}$$

where $r_1$, $r_2$, $r_F$, and $r_C$ are random numbers. $F_i$ is set to 0.1 and $F_i$ is set to 0.9. $T_1$ and $T_2$ were set to 0.1.

The above discussed adaptation strategies of $F$ and $C$ are proven to be working better than classical adaptation strategies. They add only little complexity to the algorithm, because it does the changes only in the parametric representation.

C. Deriving from History or Pool

It is also interesting that the future values for the control parameters of any algorithm is decided based on the performance of the algorithm with the past values of the control parameters. There are number of research works reported in DE literature too, in this direction. As well as, all the possible values for each of the control parameters are pooled and the algorithm is allowed to choose the required values based on the present performance of the algorithm.

This section is to discuss the adaptation strategies deriving values for the control parameter from the performance history of the algorithm or from corresponding pool of values.

The adaptive DE proposed by Hui-rongetal [45], used previous learning experiences to choose the values for the control parameters. The values of $F$ and $C$ are found at each generation. A random number ($r_n$) with uniform distribution is generated and

\[
\begin{align*}
\text{If } r_n < 0.2 & \quad F_{i_{1:1}} = F_i + \text{rand} \times (F_i - F) \\
\text{Else} & \quad C_{R_{i_{1:1}}} = C_{R_i} + \text{rand} \times (C_{R_i} - C_{R_0}) \\
\end{align*}
\]

In SHADE proposed by Ryoji Tanabe and Alex Fukunaga [46], the history information about the successful parameter values are maintained to guide DE search. It has a memory to store $H$ values of $F$ and $C$. Then the values for $F_i$ and $C_n$ are selected from the range $[1, H]$ with a random index $i$.

Another success history based model known as DESPA was proposed by Noor Awad et al, in 2015 [47]. Along with $F$ and $C$, the NP value also adapted in DESPA. Every individual $i$ in the population is assigned with its own greediness factor $p_i$. A 2-D memory structure stored with the mean values of $F$ ($\mu_F$) and $C$ ($\mu_C$) is used for control parameter adaptation. The size of memory ($M$) is set as half of $NP$. A random index $n \in [0,1]$ is chosen and is used for find $F$ and $C$.

\[
\begin{align*}
F_i &= \text{rand} \times (\mu_F, 0.1) \\
C_{R_i} &= \text{rand} \times (\mu_C, 0.1)
\end{align*}
\]

The EPSDE proposed by R Mallipeddi et al used the concept of pooled values [48]. It consists of a pool of mutation strategies and pools for corresponding parameters for those strategies. The $F$ and $C$ values are taken from a pool which has values within the range $[0.1, 0.9]$ and $[0.4, 0.9]$, respectively. Every step changes the values by 0.1. In EPSDE, each individual in the population is associated with a random mutation strategy taken from the pool. Along with the matched mutation strategy, the corresponding $F$ and $C$ values are also chosen. These values and strategies will survive until the target vector performs poorer when compared with the trial vector. Once this condition is failed, a new mutation strategy will be associated with the target vector. These strategies could be selected from the pool or from the successful combination stored before.

The CoDE [49] algorithm was also introduced in 2011 by Y Wang et al. The parameter values are predefined here. Based on the carefully selected three mutation strategies, the parameter values will be changed. During each generation a set of three trial vectors are generated. On comparison with the target vector, the best of the four (3 trial and 1 target) will go to the next generation. Three combinations are made for $F$ and $C$, based on the mutation strategies. It will follow either of the three defined as $[F=1.0, C=0.1]$, $[F=1.0, C=0.9]$ and $[F=0.8, C=0.2]$.

Wenyin Gong et al. presented a variant of JADE, called $R_n$-JADE in [50], that repairs $C_n$ based on Success history of $C_n$ values. $C_n$ and $F_n$ are selected from Normal Distribution $N(p_{CR}, 0.1)$ and Cauchy Distribution $C(u_{CR}, 0.1)$ respectively for $i^{th}$ candidate. Then $CR_i$ is repaired as per the equation given below

\[
C_{R_i} = \frac{C_{R_i} - \mu_C}{\sigma_C}
\]

where $m$ is the number of genes that were copied from mutant vector and $D$ is the problem dimension. If trial vector $U_i$ is better than target vector $X_i$, then $C_{R_i}$ and $F_{R_i}$ are added to lists of successful Crossover and Mutation Parameters ($S_{CR}$ and $S_{F}$). Then $u_{CR}$ is updated as the arithmetic mean of all the $C_i$ values in $S_{CR}$. Similarly $\mu_{CR}$ is updated as the Lehmer mean of all the $F$ values in $S_{F}$.

Comparing to other three categories, this category covers very less works in the literature. This is because of the complexity involved in remembering required information from sufficient past time and choosing the values to be achieved in the pool.

D. Parameter Adaptation with Added Logic

Another commonly used strategy for DE control parameter adaptation is to insert an additional component (or algorithm or logic) to the structure of DE. This added component will monitor DE’s performance in solving the given problem and suitably adapt the required parameters. The researchers have used some other existing algorithm as the component or have designed their own algorithm. The former one is termed in other words as hybridization of DE with other algorithms.

An algorithm hybridizing DE and Fuzzy Logic, named as FADE (Fuzzy Adaptive Differential Evolution) [51], was proposed by Lampinen and Liu in the year 2002. In FADE, a
fuzzy logic controller was used to find the values for $F$ and $C_r$ for the candidate $i$ ($F_i$ and $C_{r,i}$). The FLC-MODE (Fuzzy Logic Controlled Multi-objective Differential Evolution) [52] was introduced by FengXue et al in the year 2005, as similar to FADE.

An Iterative Function System Based Adaptive DE was proposed by Ya-Liang Li, et. al. [53]. In this algorithm the control parameters are adapted using an iterative function system. The $F_{i,G}$ and $C_{r,G}$ values are adapted using the following equations

$$F_{i,G} = \begin{cases} \alpha_i F_{i-1} + (1-\alpha_i) a_i, & \text{if } \left( f(x_{gbest}) < f(x_{i,G}) \right) \\ \text{otherwise} & \end{cases}$$

$$C_{r,G} = \begin{cases} \alpha_i C_{r,i-1} + (1-\alpha_i) a_i, & \text{if } \left( f(x_{gbest}) < f(x_{r,i}) \right) \\ \text{otherwise} & \end{cases}$$

(60) (61)

Here, $\alpha_i$, $a_i$, $\alpha_i$, and $a_i$ are uniformly generated within the range $(0,1)$. The parameter $\alpha_i$ is set to 0.5 and $a_i$ is set to 1.

Patricia Ochoa et al proposed FDE (Fuzzy Differential Evolution), which uses concept of fuzzy system for parameter adaptation [54]. The fuzzy system added to DE will give the best possible values for the control parameters. The fuzzy system has 3 membership functions ($FN_1$, $FN_2$ and $FN_3$) to mean the low, medium and high values of the parameters. It also used 3 fuzzy rules to update the values of the control parameters.

In 2009, M.G. Epitropakis et al. introduced an evolutionary approach towards self-adapting DE, known as ESADE [55]. In ESADE, a unique strategy was followed in finding the values of the control parameters. It uses two DE algorithms, one is to find the mutation rate ($F$), and the other for optimizing the given objective function. In the first DE algorithm to find $F$ value, a one-dimensional population is initialized as follows,

$$x_{g} = \{ F_{g} \}$$

(62)

where $F_{g}$ corresponds to possible values of $F$. Rather than initializing it with values in the range $(0.1, 1.0)$, based on their study they have initialized the population with values from a normal distribution with mean 0.5 and standard deviation 0.3. Once, the population has been initialized in the first DE, one generation of the second algorithm is performed. Here the fitness value of the best candidate ($f(x_{gbest})$) is taken and it is considered as the fitness value of corresponding individual of the first algorithm. For adapting $C_r$, a normal distribution with mean 0.6 and standard deviation 0.1 is considered, and values are taken from this normal distribution at every generation. Thus, in EPSADE, the first algorithm gives the Scaling factor value and using this value the second DE algorithm optimizes the given objective function.

Pravakar Roy et al proposed Differential Evolution that is Genetically Programmed [56] which ensures a self-adaptive mechanism in the DE algorithm. Here, the initial preparations are made in such a way that the need of $F$ is null. The system finds out the best crossover rate as follows, for each individual in the population of GP, a $C_r$ value is also associated with it and it is updated during the natural evolution process of GP. Initially it is taken from a Gaussian distribution and later the $GP$ will alter the values based on the predetermined fitness value. Also a counter is kept for the number of times the alteration has performed.

The adaptation strategies discussed in this section have used additional components to tune the values for the control parameters.

V. F and CR ADAPTATION STRATEGIES - INSIGHT

The research works focusing on control parameter adaptation of DE algorithm are grouped in to four categories and presented in Table 1. Due to large number of reports available in the literature, this paper aimed to consider the research works for adapting the parameters $F$ and $C_r$.

<table>
<thead>
<tr>
<th>Categories</th>
<th>I Classical Approaches</th>
<th>II Encoding of Parameters</th>
<th>III Deriving from History/Pool</th>
<th>IV With Added Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[5],[6],[7], [8],[9],[10], [11],[12],[13], [14],[15],[16], [17],[18],[19],[20],[21],[22],[23],[24],[25], [26]</td>
<td>[27],[28],[29], [30],[31],[32], [33],[34],[35], [36],[37],[38],[39],[40],[41],[42], [43],[44]</td>
<td>[45],[46],[47], [48],[49],[50]</td>
<td>[51],[52], [53],[54], [55],[56]</td>
<td></td>
</tr>
</tbody>
</table>

The research works in Category I use author defined equations to calculate the values for the parameters. Many authors also have considered using statistical distribution to select the values for the control parameters. In category II, the algorithms which encode the control parameters along with other parameters of the candidates are considered. Evolution of those parameters is done by normal DE process or by some other newly added algorithm. Recording the history of behaviour of DE in previous generations and deriving necessary information from them to decide the control parameter values for the forthcoming generations is another strategy for parameter adaptation. Research works using this strategy are grouped under this category III. Finally, in Category IV the works which consider to add additional component to DE for parameter adaptation are grouped.

VI. CONCLUSIONS

The critical parameters of DE algorithm are $F$, $C_r$ and $NP$. Selecting suitable values for them are very important as well as crucial for successful application of DE for any optimization problem. There exists no standard method for choosing values for these parameters. However, to alleviate this many parameter adaptation strategies are proposed in the literature. The existing adaptation strategies are identified and are categorized in to four groups, and brief insight about each of the identified strategies is presented in this paper.

The categories of adaptation strategies presented in this paper are strategies with classical approaches, strategies with
encoding of parameters, strategies using history/pool and strategies adding new components.

REFERENCES


