Parameter Control Mechanisms in Differential Evolution: A Tutorial Review and Taxonomy

Tsung-Che Chiang
Department of Computer Science and
Information Engineering,
National Taiwan Normal University
Taipei, Taiwan, R.O.C.
tcchiang@ieee.org

Cheng-Nan Chen
Department of Computer Science and
Information Engineering,
National Taiwan Normal University
Taipei, Taiwan, R.O.C.
698470219@ntnu.edu.tw

Yu-Chieh Lin
Department of Computer Science and
Information Engineering,
National Taiwan Normal University
Taipei, Taiwan, R.O.C.
bbeennlin@hotmail.com

(Presented in IEEE Symposium Series on Computational Intelligence (SSCI) 2013)

***Abstract*—Differential evolution (DE) is a promising algorithm for continuous optimization. Its two parameters, *CR* and *F*, have great effect on the algorithm performance. In recent years many DE algorithms with parameter control mechanisms were proposed. In this paper we propose a taxonomy to classify these algorithms according to the number of candidate parameter values, the number of parameter values used in a single generation, and the source of considered information. We classify twenty-three recent studies into nine categories and review their design features. Two types of relationships between these algorithms and several research directions are also summarized.**

***Index Terms*—differential evolution, parameter control, adaptive, self-adaptive, classification, taxonomy**

I. Introduction

Differential evolution (DE) [1][2] has been recognized as a promising algorithm for continuous optimization in the last decade. It is featured by using the difference between individuals in the mutation operator and the local selection by comparing one parent and its offspring to determine the survivor.

TABLE I. PSEUDO CODE OF DIFFERENTIAL EVOLUTION

```
$NP$: population size    $G$: generation number   $D$: problem dimension   $CR$: crossover rate    $F$: scaling factor

Initialize the population. $G = 1$.
while the stopping criterion is not met
  for $i$ = 1 to $NP$   // for each target vector $X_{i,G} = \{x_{1,i,G}, x_{2,i,G}, \ldots x_{D,i,G}\}$
    // mutation: generate a donor vector $V_{i,G} = \{v_{1,i,G}, v_{2,i,G}, \ldots v_{D,i,G}\}$
    $V_{i,G} = X_{ri1,G} + F \cdot (X_{ri2,G} - X_{ri3,G})$
    // crossover: generate a trial vector $U_{i,G} = \{u_{1,i,G}, u_{2,i,G}, \ldots u_{D,i,G}\}$
    for $j$ = 1 to $D$
      $u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if } U_j(0,1) \le CR \vee j = j_{rnd} \\ x_{j,i,G}, & \text{otherwise} \end{cases}$
    end for
    // selection: accept the trial vector if not worse than the target vector
    $X_{i,G+1} = \begin{cases} U_{i,G}, & \text{if } f(U_{i,G}) \le f(X_{i,G}) \\ X_{i,G}, & \text{otherwise} \end{cases}$
  end for
  $G = G + 1$
end while
```

Table I gives the pseudo code of a typical DE algorithm. In each generation *G*, every individual serves as the target vector. In the mutation step, several non-identical individuals are chosen randomly. One individual is the base vector and then adds an amplified difference vector to be a donor vector. Next, a trial vector is produced by taking the gene values from the target vector or the donor vector probabilistically. Finally, the trial vector replaces the target vector if the former is not worse than the latter. The DE in Table I is denoted by *rnd*/1/*bin*. Common variants include *best*/1/*bin*, *rnd*/2/*bin*, and so on [2].

DE has three parameters, *NP*, *CR*, and *F*. Experimental results have shown that their values have great effect on the convergence speed and solution quality. Setting parameter values for evolutionary algorithms is carried out in two ways: the *parameter tuning* method tests different values and runs the algorithm with the best (and fixed) value; the *parameter control* method adjusts the parameter values during the execution of algorithm. Although many advices were given for parameter tuning, it is still a time-consuming process to find proper parameter values. Moreover, we may need different parameter values for different stages in the evolution process, different individuals (search regions), and even different objectives (in the case of multiobjective optimization).

Eiben et al. [3] classified parameter control methods into three groups: deterministic, adaptive, and self-adaptive. The difference between the first two methods is in that the adaptive method considers feedback information during the evolution process. The self-adaptive method is featured by encoding parameters on the chromosomes and evolving the parameters in the same way of evolving the decision variables. After reviewing recent DE algorithms with parameter control, we found that most algorithms fall into the same group (the adaptive group) according to the classification scheme in [3]. This motivates us to propose a new taxonomy and notation to identify the features of these parameter control methods and to know the similarity and difference between them.

The rest of this paper is organized as follows. In Section II we describe the proposed taxonomy and classification criteria. Section III reviews nine categories of parameter control mechanisms in twenty-three DE studies. Section IV summarizes relationships of algorithm design and performance comparison among these algorithms. Conclusions and research directions are given in Section V.

II. Proposed Taxonomy

Although some studies addressed the dynamic control of the population size, most studies fixed the population size and focused on the control of the other two parameters, *CR* and *F*. In this paper we only consider the studies of DE that control the values of *CR* and *F*. We propose to distinguish the parameter control mechanisms by three aspects: (1) the number of candidate parameter values, (2) the number of parameter values used in a single generation, and (3) the source of considered information. The different designs and corresponding notations in the proposed taxonomy are detailed in the following.

1) The number of candidate parameter values: Almost all existing parameter control mechanisms allow any value in a predefined range, e.g. [0, 1], for *CR*. In our survey, only one study selected from a finite set of values for parameters. We denote these two kinds of strategies by *con* (continuous) and *dis* (discrete).

2) The number of parameter values used in a single generation: In this aspect, we identify four kinds of strategies in the literature.

a) 1: This is the simplest strategy. All offspring are produced by the same parameter value in a generation.

b) mul (multiple): This kind of strategy draws a random value from a specified distribution every time an offspring is produced. For example, it may draw a value for the parameter *F* from a normal distribution to generate one offspring and draw another value for another offspring.

c) idv (individual): This could be the most popular strategy. It associates with each individual one parameter value. When an individual *i* serves as the target vector, its F_i and CR_i will be used to generate the donor and trial vectors.

d) var (variable): It is like the *idv* strategy, but the parameter values are associated with the decision variables, not with individuals. We found one study taking this kind of strategy.

3) The source of considered information: When the parameter value is adjusted, information can be collected from different sources. We classify them into four groups.

a) rnd (random): This kind of strategy selects parameter values from random distributions such as the uniform distribution, normal distribution, and Cauchy distribution.

b) pop (population): It considers the statistics collected from the entire population. Common statistics include the population diversity and the successful rate of generating better offspring.

c) par (parent): This strategy is used together with the *idv* strategy in the second aspect. It adjusts the parameter values according to the parameter values of the selected parents for generating the donor and trial vectors.

d) idv (individual): This strategy is also used together with the *idv* strategy in the second aspect. It adjusts the parameter values based on the records of historical values of the target vector.

In the literature of DE, a standard three-field notation has been commonly adopted to describe the mutation strategy. For example, the *rnd/1/bin* strategy refers to that (1) the base vector is selected randomly, (2) one difference vector is used in generating a donor vector, and (3) the binomial crossover is used to produce the trial vector. Similarly, we propose a three-field notation to give a simple and pertinent tag to the parameter control mechanisms. For example, the *con/mul/pop* strategy refers to that (1) parameter values are from a continuous range, (2) multiple values are used in a single generation, and (3) parameter values are adjusted based on the statistics collected from the entire population.

III. Literature Review

We classify the literature on the parameter control of DE into nine groups. For each group, we will give some examples and describe their core design ideas.

A. con/1/pop

Ali and Tőrn [4] proposed the DEPD, in which the value of *CR* is fixed and the value of *F* is adjusted by (1). $F_{\min}$ denotes the minimum value of *F*, and $f_{\max}/f_{\min}$ denotes the maximum/minimum fitness value in the population. When the difference between the fitness values of the best and the worst individuals decreases, the value of *F* increases. It follows the common idea that a larger perturbation is made when the population diversity gets lower.

$$F = \begin{cases} \max\{F_{\min}, 1 - \left| f_{\max} / f_{\min} \right| \}, & \text{if } \left| f_{\max} / f_{\min} \right| < 1, \\ \max\{F_{\min}, 1 - \left| f_{\min} / f_{\max} \right| \}, & \text{otherwise.} \end{cases} \tag{1}$$

Liu and Lampinen proposed to use two fuzzy logic controllers (FLCs) to adjust *CR* and *F* in their FADE [5]. The inputs of the FLC are the change of values of decision variables (d_1) and the change of objective values (d_2) between two generations. When d_1 is small, *CR* and *F* increase as d_2 increases. When d_2 is medium or large, *CR* and *F* increase as d_1 increases.

The ADEA [6] was proposed by Qian and Li to deal with multiobjective optimization problems. It follows the well-known NSGA-II [7] to separate the individuals into different fronts and calculates a similar crowding measure. The value of *F* is adjusted according to how well the individuals are evenly distributed on the fronts and how many individuals are non-dominated solutions. The detailed equation is expressed in (2). Assume that there are *k* fronts and there are m_j individuals in the j^{th} front. The symbol d_{ij} is the crowding measure of an individual *i* in the j^{th} front, $\overline{d_j}$ is the average crowding measure in the j^{th} front, $\overline{d}$ is the average crowding measure of all individuals, and *df* is the Euclidean distance between the two boundary solutions. |*P*| and |*Q*| denote the number of non-dominated solutions and the population size, respectively. Generally speaking, the value of *F* increases when the individuals are not evenly distributed and when the number of non-dominated solutions is small.

$$F = \max\left(\frac{\sum_{j=1}^{k} \sum_{i=1}^{m_j} \left| d_{ij} - \overline{d_j} \right| + df}{\sum \left| Q \right| \cdot \overline{d} + df}, 1 - \frac{2\left| P \right|}{\left| Q \right|}, F_{\min} \right) \tag{2}$$

B. con/mul/rnd

Yang et al. [8] focused on the control of the parameter F. They noticed that the normal distribution $N(0, 1)$ is likely to produce small values and the Cauchy distribution has a greater probability of producing larger values. Thus, their NSDE adjusts the value of F by the two distributions in equal probability, as shown in (3). The mean and standard deviation of the normal distribution were taken after determining an empirical value for F. In (3), δ is a Cauchy random variable with scale parameter $t = 1$. The value of F changes every time a donor vector is produced.

$$F_i = \begin{cases} N(0.5, 0.5), & \text{if } U_i(0,1) < 0.5, \\ \delta, & \text{otherwise.} \end{cases} \tag{3}$$

C. con/mul/pop

The *con/mul/pop* strategy is extended from the *con/mul/rnd* strategy. It also draws the values of CR and F from specified random distributions. However, it does not use fixed values for the distribution parameters (e.g. the mean of the normal distribution). Instead, it adjusts the values of the distribution parameters based on the population statistics.

The SaDE [9][10] has adaptive control of the mutation strategies and the value of CR. Given K mutation strategies, the algorithm calculates the probability $p_{k,g}$ of choosing a strategy k at generation g following (4) and (5). Simply speaking, it counts the number of successful trials $ns_{k,j}$ and the number of failed trials $nf_{k,j}$ of producing better offspring for each strategy k every LP generations. Then, the successful rate $S_{k,g}$ is calculated, and each strategy is selected in a probability proportional to its successful rate. The additional parameter LP determines the duration of collecting statistics. Another additional parameter ε was introduced to give small chances to the strategies with zero successful rates.

$$S_{k,g} = \frac{\sum_{j=g-LP}^{g-1} ns_{k,j}}{\sum_{j=g-LP}^{g-1} ns_{k,j} + \sum_{j=g-LP}^{g-1} nf_{k,j}} + \varepsilon \tag{4}$$

$$p_{k,g} = \frac{S_{k,g}}{\sum_{j=1}^{K} S_{j,g}} \tag{5}$$

Control of the parameter CR is done in a similar way. The values of CR that lead to better offspring are recorded for each mutation strategy k. Then, the median CRm_k of these successful CR values is taken as the mean of the normal distribution when the mutation strategy k is chosen.

$$CR_{k,i} = N(CRm_k, 0.1) \tag{6}$$

Yang et al. [11] proposed the SaNSDE, which was derived from SaDE and NSDE. It uses the same mechanism of selecting among multiple mutation strategies as SaDE does. As for CR, it is like SaDE and considers more detailed information when calculating CRm. In addition to recording the successful CR values in the set CR_{rec}, it also records the improvement on fitness of these successful CR values in Δf_{rec}. The value of CRm is the weighted average of all successful CR values, where the weight w_j is the portion of the total improvement contributed by the j^{th} CR value.

$$CRm = \sum_{j=1}^{|CR_{rec}|} w_j \cdot CR_{rec}(j) \tag{7}$$

$$w_j = \Delta f_{rec}(j) \Big/ \left(\sum_{c=1}^{|\Delta f_{rec}|} \Delta f_{rec}(c) \right) \tag{8}$$

To control the value of *F*, SaNSDE takes the dual-distribution design from NSDE. It is enhanced by adapting the probability *fp* of selecting between the normal and Cauchy distributions based on the successful rate of these two distributions. Another small difference is that the standard deviation of the normal distribution was changed from 0.5 to 0.3.

$$F_i = \begin{cases} N(0.5, 0.3), & \text{if } U_i(0,1) < fp, \\ \delta, & \text{otherwise.} \end{cases} \tag{9}$$

The JADE proposed by Zhang and Sanderson [12] is another good representative of the *con/mul/pop* strategy. It adjusts the values of *CR* and *F* in the way similar to the method of controlling *CR* in SaDE. One difference is that JADE updates the distribution parameters progressively at each generation. As in (11), the new mean μ_{CR} of the normal distribution in (10) is the weighted sum of the current mean and the arithmetic average of the successful *CR* values in CR_{rec}. The method of adjusting *F* is the same except that the normal distribution and arithmetic mean are replaced by the Cauchy distribution and Lehmer mean, as described in (12)-(14). The use of the Cauchy distribution and the Lehmer mean is to place more chances on larger *F* values. The JADE2 [13] followed JADE and used the same parameter control methods. It dealt with multiobjective optimization problems.

$$CR_i = N(\mu_{CR}, 0.1) \tag{10}$$

$$\mu_{CR} = (1-c)\cdot\mu_{CR} + c\cdot mean_A(CR_{rec}) \tag{11}$$

$$F_i = C(\mu_F, 0.1) \tag{12}$$

$$\mu_F = (1-c)\cdot\mu_F + c\cdot mean_L(F_{rec}) \tag{13}$$

$$mean_L(F_{rec}) = \sum_{F\in F_{rec}} F^2 \Big/ \sum_{F\in F_{rec}} F \tag{14}$$

Gong et al. [14] followed JADE and proposed the SaJADE. It utilizes the idea in JADE to control the selection among *K* mutation strategies. In (16) μ_s is set by 0.5 and the standard deviation is set by 1/6 at the first generation ($g = 1$) to ensure that η_i is generated in the range [0, 1). After starting the collection of successful mutation strategies in S_{rec}, the standard deviation is set smaller (0.1) to emphasize the adaptation effect.

$$S_i = \lfloor \eta_i \cdot K \rfloor + 1 \tag{15}$$

$$\eta_i = \begin{cases} N(\mu_s, 1/6), & \text{if } g = 1 \\ N(\mu_s, 0.1), & \text{otherwise} \end{cases} \tag{16}$$

$$\mu_s = (1-c)\cdot\mu_s + c\cdot mean_A(S_{rec}) \tag{17}$$

D. con/idv/rnd

Similar to the *con*/*mul*/*rnd* strategy, the *con*/*idv*/*rnd* strategy adjusts the parameter values based on specified probability distributions such as uniform distribution. The difference is in that the *con*/*idv*/*rnd* strategy records the parameter values on the individuals so that the selection procedure of the evolution process can help to identify good parameter values.

The jDE proposed by Brest et al. [15] leads the studies in this category. Before an offspring is produced for a target vector *i* at generation (*g*+1), there is a probability (τ_1 and τ_2 in (18) and (19)) of changing the values of *F* and *CR* to a random value within the predetermined range. In their experiments, the authors proposed to give a small probability of change and set τ_1 and τ_2 both by 0.1.

$$F_{i,g+1} = \begin{cases} U_1(F_{\min}, F_{\max}), & \text{if } U_2(0,1) < \tau_1, \\ F_{i,g}, & \text{otherwise.} \end{cases} \tag{18}$$

$$CR_{i,g+1} = \begin{cases} U_3(0,1), & \text{if } U_4(0,1) < \tau_2, \\ CR_{i,g}, & \text{otherwise.} \end{cases} \tag{19}$$

Later, Brest et al. proposed jDE-2 [16] by integrating the idea of multiple mutation strategies in SaDE into jDE. It records the values of *F* and *CR* for each of the three adopted mutation strategies. In addition, jDE-2 replaces the *k* worst individuals at every *l* generation with parameter values randomly selected from the feasible range. This may speed up the adaptation of parameter values.

Soliman and Bui [17] proposed a control strategy like that in jDE but added more randomness in the control of *F*. Instead of controlling the value of *F* directly, their strategy samples the *F* value based on the Cauchy distribution and adjusts the scale parameter of the Cauchy distribution in a small probability (τ_1). The value of *CR* is controlled in the same way as in jDE. Besides τ_1 and τ_2, three more parameters, μ, δ_l, and δ_u are required. The authors did not name their strategy, and in the following we call it CSDE.

$$F_{i,g+1} = \begin{cases} C(\mu, \delta_{i,g+1}), & \text{if } U_1(0,1) < \tau_1, \\ C(\mu, \delta_{i,g}), & \text{otherwise.} \end{cases} \tag{20}$$

$$\delta_{i,g+1} = \delta_l + \delta_u \cdot U_2(0,1) \tag{21}$$

The strategy in MOSADE [18] can be viewed as a special case of that in jDE with τ_1 and τ_2 set by 1. In other words, the values of *F* and *CR* are re-sampled at every generation.

E. con/idv/pop

The *con*/*idv*/*pop* strategy records parameter values on the individuals and adjusts the values using the information on the target vector as well as the whole population. The RADE [19] calculates the accumulated fitness improvement Δf_i for each individual *i* every α generations. The individuals whose Δf_i is among the top $1/\beta$% keep their *F* values in the next α generations. The remaining individuals re-sample a random value within the feasible range [F_{min}, F_{max}]. The values of α and β were set by 5 and 2, respectively. The value of *CR* was a constant depending on the problem dimension.

$$F_{i,g+\alpha} = \begin{cases} F_{i,g}, & \text{if } \Delta f_i \text{ is among the top } 1/\beta\%, \\ U(F_{\min}, F_{\max}), & \text{otherwise.} \end{cases} \tag{22}$$

Jia et al. [20] proposed the ISADE as an extension of the jDE. When the parameter value on an individual X_i is to be changed, ISADE has two options. If the fitness $f(X_i)$ is smaller than the average fitness f_{avg} over the whole population, the value changes toward F_{min}. The better the individual is, the smaller the value of *F* is. The rationale behind is to search locally for good individuals. If the fitness is equal to or greater than the average fitness, a random value within the specified range is chosen. The value of *CR* is controlled in exactly the same way, and thus the equation is omitted here.

$$F_i = \begin{cases} \begin{cases} F_{\min} + (F_i - F_{\min}) \cdot \dfrac{f(X_i) - f_{\min}}{f_{avg} - f_{\min}}, & \text{if } f(X_i) < f_{avg}, \\ U_2(F_{\min}, F_{\max}), & \text{if } f(X_i) \geq f_{avg} \end{cases} & \text{if } U_1(0,1) < \tau_1 \\ F_i, & \text{otherwise} \end{cases} \tag{23}$$

F. con/idv/par

The *con*/*idv*/*par* strategy evolves the parameter on the individuals in the same way as it evolves the decision variables. In other words, the parameter values are adjusted based on the information on the parents. This category matches the “self-adaptive” category in Eiben et al. [3]. (In fact, most of the so-called self-adaptive DE algorithms in the literature fall into the “adaptive” category according to their taxonomy. Our taxonomy helps to further identify their features.)

The SPDE by Abbass [21] adjusts the value of *CR* of a target vector *i* according to the values of *CR* of the three randomly selected parents r_1, r_2, and r_3, as (24) defines.

$$CR_i = CR_{r1} + N(0,1) \cdot (CR_{r2} - CR_{r3}) \tag{24}$$

Omran et al. [22] proposed the SDE, which adjusted the value of *F* by (25). Note that the individuals used to adjust the *F* value are different from the individuals used to adjust the decision variables.

$$F_i = F_{r4} + N(0,0.5) \cdot (F_{r5} - F_{r6}) \tag{25}$$

Instead of using the normal distribution in the SPDE and SDE, the DESAP [23] uses the value of scaling factor *F* in the adjustment of *CR* in (28). In the experiments, the value of *F* was fixed as one.

$$CR_i = CR_{r1} + F \cdot (CR_{r2} - CR_{r3}) \tag{26}$$

Zamuda et al. [24] proposed the DEMOwSA. The new value of *CR* is the product of a random variable $e^{\tau N(0,\,1)}$ and the average *CR* over the target vector and three parents. The value of the extra parameter τ was set by 1/(8$\sqrt{2D}$), *D* being the problem dimension. The value of *F* is adjusted in the same way, and the equation is omitted.

$$CR_i = \frac{CR_i + CR_{r1} + CR_{r2} + CR_{r3}}{4} \cdot e^{\tau \cdot N(0,1)} \tag{27}$$

G. con/idv/idv

As the name indicates, the *con*/*idv*/*idv* strategy adjusts the parameter values recorded on the individual based on the individual's own information. The SFLSDE [25] is a descendant of the jDE. It is different from the jDE in that two local searches are carried out probabilistically to search for the proper *F* value for the best individual in the population. (The control mechanism for *F* of the remaining individuals and the control mechanism for *CR* of all individuals are identical to those of the jDE.) Given the best individual and its current *F* value, the local search procedure repeats generating new *F* values, using these *F* values to produce new offspring individuals, and accepting the *F* value that leads to the best individual. In (28), the golden section search and hill climbing search use different neighborhood functions to generate the new *F* values. Details are referred to [25].

$$F_{i,g+1} = \begin{cases} \text{Golden section search,} & \text{if } U_1(0,1) < \tau_1, \\ \text{Hill climbing search,} & \text{if } \tau_1 \le U_1(0,1) < \tau_2, \\ \begin{cases} U_3(F_{\min}, F_{\max})\cdot, \text{ if } U_2(0,1) < \tau_3, \\ F_{i,g} \qquad \text{otherwise} \end{cases} & \text{if } \tau_2 < U_1(0,1) \end{cases} \tag{28}$$

Pan et al. [26] proposed the SspDE, in which each individual *i* has its own lists CRL_i, FL_i, and SL_i, recording *CR* values, *F* values, and mutation strategies, respectively. Each of these lists consists of *LP* elements and is updated every *LP* generations. In each generation each individual serves as the target vector and uses its own *CR*, *F*, and mutation strategy to produce an offspring. If the offspring is not worse, record the *CR*, *F*, and mutation strategy into another three lists $wCRL_i$ (*w* for winning), wFL_i, and wSL_i, respectively. Every *LP* generations, the CRL_i list is refilled by the values in $wCRL_i$ in probability *RP* and by random values in probability (1 – *RP*). The FL_i and SL_i lists are updated in the same way. *LP* and *RP* are two additional parameters.

H. con/var/pop

The *con*/*var*/*pop* strategy is like the *con*/*idv*/*pop* strategy, but it associates parameter values with decision variables instead of individuals. The APDE [27] is an example. Based on a theoretical result [28] of the relation between the variance of values of decision variables $Var(x^i)$ and the values of CR_i and F_i, APDE adjusts F_i values in even generations by (29) and adjusts CR_i values in odd generations by (30). In (29) and (30), *N* denotes the population size, and c_i is calculated based on $Var(x^i)$ and the variance of the objective values $Var(f_j)$ among the population in the previous (g) and current (g+1) generations. The APDE deals with multiobjective optimization, and $Var(f(g))$ is the average of $Var(f_j(g))$ over M objectives. In (31), γ is an additional parameter and was set by 1.25 in the APDE.

$$F_i = \begin{cases} \sqrt{K/2N \cdot CR_i}, & \text{if } K \geq 0, \\ F_{\min}, & \text{otherwise.} \end{cases} \quad (29)$$
$$K = N(c_i - 1) + CR_i(2 - CR_i)$$

$$CR_i = \begin{cases} -(F_i^2 N - 1) + \sqrt{(F_i^2 N - 1)^2 - N(1 - c_i)}, & \text{if } c_i \geq 1 \\ CR_{\min} & \text{otherwise} \end{cases} \quad (30)$$

$$c_i(g+1) = \gamma \frac{Var(x^i(g))Var(f(g))}{Var(x^i(g+1))Var(f(g+1))}$$
$$Var(f(g)) = \frac{1}{M}\sum_{j=1}^{M} Var(f_j(g)) \quad (31)$$

I. *dis/mul/pop*

Different from the above strategies, the *dis*/*mul*/*pop* is featured by a finite set of candidate parameter values. DEBR [29] pre-specified nine combinations of parameter values by CR = {0, 0.5, 1} and F = {0.5, 0.8, 1}. Let n_h denote the number of successful trials of generating better offspring by a combination h in previous generations, the probability q_h of selecting h is defined by (32). H is the total number of combinations. In general, the probability is proportional to the number of successes. The additional parameter n_0 is introduced to avoid dramatic variation of the probability. If any probability q_h decreases below a given limit δ, it will be reset to the starting value 1/H.

$$q_h = \frac{n_h + n_0}{\sum_{j=1}^{H} n_j + n_0} \quad (32)$$

IV. Relationship

In last two sections we propose a taxonomy to identify the features of parameter control mechanisms of DE and review state-of-the-art studies in each category. The categorization helps to see the similarities and differences between the existing control mechanisms. In this section we summarize two relationships between these algorithms.

The first is the design relationship, depicted in Fig. 1. In Fig. 1, $A \leftarrow B$ means that algorithm B is derived from algorithm A. For example, SaNSDE combines the idea of different statistical distributions in NSDE and the idea of adaptive probability in SaDE. As more new algorithms will be developed based on the existing ones, the design relationship will keep growing. This relationship could help researchers to track how an algorithm evolved in the history.

Another relationship is constructed by a summary of the performance comparison results in the reviewed studies. When an algorithm A claimed that it was superior to algorithm B, we put an arrow as $A \rightarrow B$ in Fig. 2. Perhaps due to the simplicity, jDE is the most popular benchmark algorithm in the literature. Note that the testing functions and performance measures in these papers were not necessary identical (although many of them did use the same functions and measures). Another note is that performance difference between these algorithms is not totally determined by the parameter control mechanism. For example, the good performance of JADE is partially contributed by an external archive and the modified mutation strategy. On one hand, this relationship helps us to find out competitive benchmark algorithms for performance comparison. On the other hand, it reveals that more numerical studies are required to complete this relationship diagram. Neri and Tirronen [30] conducted an experimental analysis on eight advanced DE including jDE, SaDE, and SFLSDE. They reported the good performance of jDE and SFLSDE. The analysis pointed out that the original DE employs too much a deterministic search logic and may suffer from the stagnation condition. The restricted randomization in the jDE enriches the set of moves, and the local search in the SFLSDE increases the exploitative pressure within the explorative DE structure. Researches like this experimental study are very helpful for the design of parameter control mechanisms.

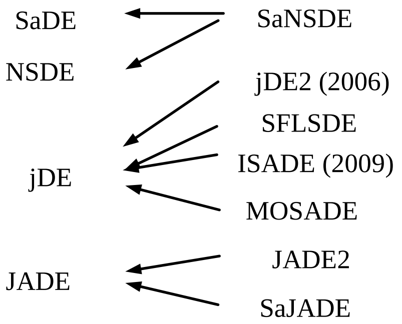

Fig. 1. Design relationship

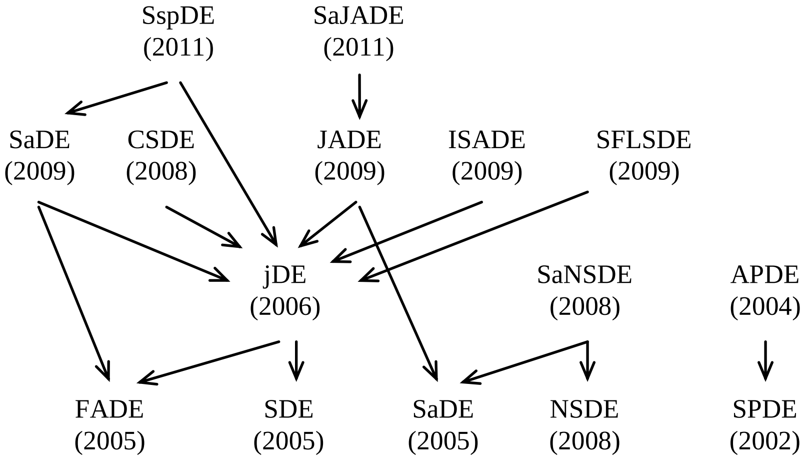

Fig. 2. Performance comparison relationship

V. Conclusions and Research Directions

In this study we reviewed literature on DE algorithms with parameter control and proposed a taxonomy and notation to classify these methods. These algorithms are different from one another mainly in the number of parameter values used in a single generation (*1*, *mul*, *idv*, and *var*) and the source of considered information (*rnd*, *pop*, *par*, and *idv*). Our three-field notation gives a quick and informative tag. Table II presents nine categories of parameter control mechanisms and representative algorithms. It also gives the additional parameters, the number of objectives in the optimization problems, and brief descriptions. There is still much to investigate. Here we give four directions for future research:

1) Making the algorithm simpler: The motivation to do parameter control is twofold. One is to adapt parameter values to different problems, search stages, search regions, etc. to improve algorithm performance. The other is to reduce users' burden on testing and finding the proper parameter setting. The new algorithms achieve better and better solution quality, but most of them replace the parameters of DE (i.e. *CR* and *F*) with another set of parameters of the proposed control mechanisms. For example, the jDE-like algorithms need to define the probability (τ) of random changes and the SaDE-like algorithms need to set the learning period (*LP*). Although the algorithm performance was shown to be not sensitive to the values of these newly introduced parameter (e.g. [10][26]), it would be better if we can save this task for the users.

2) *Considering problem-oriented information:* Various information has been considered for adjusting parameter values. In general, the information is either quality-based or diversity-based. A very common quality-based information is the (weighted) successful times/rate of producing better offspring. The relative quality of individuals to the population is sometimes used. Diversity-based information includes fitness difference between the best and worst individuals, variance of values of decision variables, and distribution along the Pareto front. All are search process-oriented. They measure how well the search process goes. We should consider also problem-oriented information. Many useful experiences have been found in the literature on parameter tuning. If we can identify the problem characterisitcs such as unimodal/multimodal and separable/non-separable, we can do parameter control more strategically.

3) Adapting with respect to multiple objectives: In our survey, the number of researches on adaptive DE for multiobjective optimization is still much fewer than that of the single-objective ones. We have seen algorithms that adapt parameter values in the individual-wise (·/*idv*/·) and variable-wise (·/*var*/·) manners. When multiple objectives are to be minimized, it is likely that different values of *CR* and *F* must be used for different objectives. We are looking forward to seeing future researches investigating this issue.

4) *Doing parameter control through distributed DE:* The ·/*1*/· and ·/*idv*/· strategies are the two extremes in terms of the control granularity. The former takes the whole population to find the good parameter values, and the latter lets every individual do by itself. Doing parameter control through several (structured) sub-populations may be a good balance. Some promising initial results can be found in [31] and [32].

REFERENCES

[1] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

[2] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," IEEE Transactions on Evolutionary Computation, vol. 15, no. 1, pp. 4–31, 2011.

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

[4] M. M. Ali and A. Tőrn, "Population set-based global optimization algorithms: some modifications and numerical results," Computers & Operations Research, vol. 31, no. 10, pp. 1703–1725, 2004. [DEPD]

[5] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," Soft Computing, vol. 9, no. 6, pp. 448–462, 2005. [FADE]

[6] W. Qian and A. Li, "Adaptive differential evolution algorithm for multiobjective optimization problems," Applied Mathematics and Computation, vol. 201, no. 1-2, pp. 431–440, 2008. [ADEA]

[7] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm：NSGA-II, " IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, 2002.

[8] Z. Yang, X. Yao, J. He, "Making a difference to differential evolution," In P. Siarry and Z. Michalewicz (Eds.) Advance in Metaheuristics for Hard Optimization, 2008, pp. 397–414. [NSDE]

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

[10] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," IEEE Transactions on Evolutionary Computation, vol. 13, no. 2, pp. 398–417, 2009. [SaDE]

[11] Z. Yang, K. Tang, and X. Yao, "Self-adaptive differential evolution with neighborhood search," IEEE Congress on Evolutionary Computation, pp. 1110–1116, 2008. [SaNSDE]

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

[13] J. Zhang and A. C. Sanderson, "Self-adaptive multi-objective differential evolution with direction information provided by archived inferior solutions," IEEE Congress on Evolutionary Computation, pp. 2801–2810, 2008. [JADE2]

[14] W. Gong, Z. Cai, C. X. Ling, and H. Li, "Enhanced differential evolution with adaptive strategies for numerical optimization," IEEE Transactions on Systems, Man, and Cybernetics – Part B, vol. 41, no. 2, pp. 397–413, 2011. [SaJADE]

[15] 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. [jDE]

[16] J. Brest, V. Žumer, and M. S. Maučec, "Self-adaptive differential evolution algorithm in constrained real-parameter optimization," IEEE Congress on Evolutionary Computation, pp. 215 –222, 2006. [jDE-2]

[17] O. S. Soliman and L. T. Bui, "A self-adaptive strategy for controlling parameters in differential evolution," IEEE Congress on Evolutionary Computation, pp. 2837–2842, 2008. [CSDE]

[18] Y. N. Wang, L. H. Wu, and X. F. Yuan, "Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure," Soft Computing, vol. 14, no. 3, pp. 193–209, 2010. [MOSADE]

[19] A. Nobakhti and H. Wang, “A simple self-adaptive differential evolution algorithm with application on the ALSTOM gasifier,” Applied Soft Computing, vol. 8, no. 1, pp. 350–370, 2008. [RADE]

[20] L. Jia, W. Gong, and H. Wu, “An improved self-adaptive control parameter of differential evolution for global optimization,” In Z. Cai et al. (Eds.) Computational Intelligence and Intelligent Systems, 2009, pp. 215–224. [ISADE]

[21] H. A. Abbass, “The self-adaptive Pareto differential evolution algorithm,” IEEE Congress on Evolutionary Computation, pp. 831–836, 2002. [SPDE]

[22] M. G. H. Omran, A. Salman, and A. P. Engelbrecht, “Self-adaptive differential evolution,” Lecture Notes in Artificial Intelligence, vol. 3801, pp. 192–199, 2005. [SDE]

[23] J. Teo, “Exploring dynamic self-adaptive populations in differential evolution,” Soft Computing, vol. 10, pp. 673–686, 2006. [DESAP]

[24] A. Zamuda, J. Brest, B. Bošković, and V. Žumer, “Differential evolution for multiobjective optimization with self adaptation,” IEEE Congress on Evolutionary Computation, pp. 3617–3624, 2007. [DEMOwSA]

[25] F. Neri and V. Tirronen, “Scale factor local search in differential evolution,” Memetic Computing, vol. 1, no. 2, pp. 153–171, 2009. [SFLSDE]

[26] Q. K. Pan, P. N. Suganthan, L. Wang, L. Gao, and R. Mallipeddi, “A differential evolution algorithm with self-adapting strategy and control parameters,” Computers & Operations Research, vol. 38, no. 1, pp. 394–408, 2011. [SspDE]

[27] D. Zaharie and D. Petcu, “Adaptive Pareto differential evolution and its parallelization,” Lecture Notes in Computer Science, vol. 3019, pp. 261–268, 2004. [APDE]

[28] D. Zaharie, “Control of population diversity and adaptation in differential evolution algorithms,” International Conference on Soft Computing, pp. 41–46, 2002.

[29] J. Tvrdik, “Adaptation in differential evolution: a numerical comparison,” Applied Soft Computing, vol. 9, no. 3, pp. 1149–1155, 2009. [DEBR]

[30] F. Neri and V. Tirronen, Recent advances in differential evolution: a survey and experimental analysis, Artificial Intelligence Review, vol. 33, pp. 61–106, 2010.

[31] M. Weber, V. Tirronen, and F. Neri, Scale factor inheritance mechanism in distributed differential evolution, Soft Computing, vol. 14, pp. 1187–1207, 2010.

[32] M. Weber, F. Neri, and V. Tirronen, A study on scale factor in distributed differential evolution, Information Sciences, vol. 181, pp. 2488–2511, 2011.

TABLE II. PROPOSED TAXONOMY AND EXAMPLES

Category[*]	Algorithm	Adjusted parameters	Additional parameters[**]	#Obj	Brief descriptions
con/1/*pop*	DEPD 2004	*F*		SO	(1) Fixed *CR*; (2) Increase *F* when $abs(f_{\max}/f_{\min})$ decreases.
	FADE 2005	*CR*, *F*	membership functions	SO	(1) Two fuzzy logic controllers for *CR* and *F*, respectively; (2) Inputs of the controllers are based on the average genotypic and phenotypic distances between two generations.
	ADEA 2008	*F*		MO	(1) Fixed *CR*; (2) Increase *F* when the individuals are not evenly distributed and when the number of non-dominated solutions is small.
con/mul/rnd	NSDE 2008	*F*		SO	(1) CR~U(0, 1); (2) F~N(0.5, 0.5)/Cauchy in equal probability.
con/mul/pop	SaDE 2005/2009	*CR*, *s*	LP, ε	SO	(1) CR~N(CRm, 0.1), *CRm* as the median of successful *CR* values; (2) F~N(0.5, 0.3); (3) Selection probability of mutation strategies is proportional to the successful rate.
	SaNSDE 2008	*CR*, *F*, *s*	*LP*	SO	(1) Descendant of NSDE and SaDE; (2) CR~N(CRm, 0.1), *CRm* as the weighted average of successful *CR* values, where weights are the portion of improvement on fitness; (3) F~N(0.5, 0.5)/Cauchy with probability depending on the successful rate.
	JADE 2009	*CR*, *F*	*c*	SO	(1) CR~N(μ, 0.1), μ as the weighted sum of current μ and arithmetic mean of successful *CR* values; (2) F~C(μ, 0.1), μ as the weighted sum of current μ and Lehmer mean of successful *CR* values.
	JADE2 2008	*CR*, *F*	*c*	MO	(1) Descendant of JADE, using the same parameter control mechanism.
	SaJADE 2011	*CR*, *F*, *s*	*c*	SO	(1) Descendant of JADE; (2) Selection of mutation strategies by the same mechanism in JADE.
con/idv/rnd	jDE 2006	*CR*, *F*	τ_1, τ_2	SO	(1) Change *CR* by $U(0, 1)$ with probability τ_1; (2) Change *F* by $U(F_{\min}, F_{\max})$ with probability τ_2.
	jDE-2 2006	*CR*, *F*, *s*	τ_1, τ_2, k, l	SO	(1) Descendant of jDE, adding multiple mutation strategies; (2) Re-initialize the parameter values of the worst *k* individuals every *l* generations.
	CSDE 2008	*CR*, *F*	τ_1, τ_2, μ, δ_l, δ_u	SO	(1) Change *CR* by $U(0, 1)$ with probability τ_1; (2) Change *F* by $C(\mu, \delta)$ with probability τ_2, δ~$U(\delta_l, \delta_u)$.

	MOSADE 2010	*CR*, *F*		MO	(1) Descendant of jDE;
					(2) CR~U(0.0, 0.5);
					(3) F~U(0.1, 0.9).

con/idv/pop	RADE 2008	*F*	α, β	SO	(1) For the individuals whose accumulated fitness improvement is among the top 1/β% in the population, keep their *F* values; for the remaining individuals, set random values (2) Update *F* values every α generations.
	ISADE 2009	*CR*, *F*	τ_1, τ_2	SO	(1) Descendant of jDE; (2) Change the values of *CR* and *F* toward the lower bound if the individual's fitness is better than the average fitness over the population.
con/idv/par	SPDE 2002	*CR*		MO	(1) $CR_i = CR_{r1} + N(0, 1)\cdot(CR_{r2} - CR_{r3})$; (2) F~$N(0, 1)$.
	SDE 2005	*F*		SO	(1) CR~N(0.5, 0.15); (2) $F_i = F_{r4} + N(0, 0.5)\cdot(F_{r5} - F_{r6})$.
	DESAP 2006	*CR*		SO	(1) $CR_i = CR_{r1} + F\cdot(CR_{r2} - CR_{r3})$; (2) Fixed *F* value.
	DEMOwSA 2007	*CR*, *F*	τ	MO	(1) $CR_i = (1/4)\cdot(CR_i + CR_{r1} + CR_{r2} + CR_{r3})\cdot e^{\tau\cdot N(0, 1)}$; (2) $F_i = (1/4)\cdot(F_i + F_{r1} + F_{r2} + F_{r3})\cdot e^{\tau\cdot N(0, 1)}$.
con/idv/idv	SFLSDE 2009	*CR*, *F*	τ_1, τ_2, τ_3, τ_4	SO	(1) Descendant of jDE; (2) Use the golden section search and hill climbing search probabilistically to adjust the *F* value of the best individual.
	SspDE 2011	*CR*, *F*, *s*	*LP*, *RP*	SO	(1) Each individual has its own lists, CRL_i, FL_i, and SL_i; (2) Successful *CR*, *F*, and strategy are recorded in $wCRL_i$, wFL_i, and wSL_i; (3) Every *LP* generations, CRL_i, FL_i, and SL_i are refilled by $wCRL_i$, wFL_i, and wSL_i, respectively; (4) Refilled values are taken from the winning lists in probability *RP*.
con/var/pop	APDE 2004	*CR*, *F*	γ	MO	(1) Adjust the parameter values to maintain the variances of decision variables and average variance of objective values between two generations.
dis/mul/pop	DEBR 2009	*CR*, *F*, *s*	n_0, δ	SO	(1) Use a prespecified combinations of *CR*, *F*, and mutation strategies; (2) Select one combination in a probability proportional to the successful rate.

[*] Comparing with the classification scheme in [3], the *·/·/rnd* category matches the ***deterministic parameter control*** category, the *con/idv/par* category matches the ***self-adaptive parameter control*** category, and the remaining categories match the ***adaptive parameter control*** category.

[**] The upper bound and lower bound of *CR* and *F* values are not listed as additional parameters here.