RESEARCH ARTICLE
doi: 10.2306/scienceasia1513-1874.2017.43S.052
ScienceAsia 43S (2017): **52–68**

The analysis, identification and measures to remove inconsistencies from differential evolution mutation variants

Qamar Abbas[a,*], **Jamil Ahmad**[b], **Hajira Jabeen**[c]

[a] Department of Computer Science and Software Engineering, International Islamic University, Islamabad, 44000, Pakistan
[b] Vice Chancellor, Kohat University of Science and Technology (KUST), Kohat, 44000, Pakistan
[c] Computer Department, Iqra University, Islamabad, 44000, Pakistan

*Corresponding author, e-mail: qamar.abbas@iiu.edu.pk

Received 1 Dec 2014
Accepted 1 Jun 2017

ABSTRACT: Differential evolution (DE) is a powerful global optimization algorithm which has been studied intensively by many researchers in recent years. A number of mutation variants have been established for this algorithm. These mutation variants make the DE algorithm more applicable, but random development of these variants has created inconsistencies such as naming and formulation. Hence this study aims to identify inconsistencies and to propose solutions to make them consistent. Most of the inconsistencies exist because of the uncommon nomenclature used for these variants. In this study, a comprehensive study is carried out to identify inconsistencies in the nomenclature of mutation variants that do not match each other. Appropriate and consistent names are proposed for them. The proposed names assigned to conflicting variants are based on the name of the variant, the total number of vectors used to generate the trial vector, and the order of the vectors to form the equation of these mutation variants. To ensure the performance diversity of the consistent set of DE mutation strategies, experimental results are generated using a test suit of benchmark functions.

KEYWORDS: crossover, optimization, algorithm

INTRODUCTION

Differential evolution (DE), proposed by Storn and Price[1], is a stochastic-population-based evolutionary algorithm. DE is simple, easy to use, and speedy, and has a greater probability to find the global optima for any given function[2,3]. DE has been successfully used in various systems such as electrical power systems[4], microwave engineering[5], robotics[6], bioinformatics[7], chemical engineering[8], pattern recognition[9], artificial neural networks[10], and signal processing[11]. In DE algorithm, a population of potential solutions is randomly initialized within an n-dimensional search space. All potential solutions are equally likely to be selected as a parent in DE algorithm. The candidate solutions evolve themselves by exploring the entire search space over time to locate the optima of the objective function. A new vector is generated by adding the weighted difference between two population vectors to a third vector at each iteration of DE algorithm. Three vectors, randomly selected from the existing population, are used to generate each new vector. Many algorithms are used for numerical benchmark optimization, but DE has shown better performance than genetic algorithms and particle swarm optimization for such problems[1,12]. There are many parameters in DE algorithm such as the population size N_{p}, mutation probability F, and crossover C_r. DE algorithm has many mutation strategies[2] in the literature. Various state-of-the-art versions of DE algorithms such as ADE[13], jDE[2], SaDE[14], JADE[15], DEGL[16], CoDE[17], EPSDE[18], $\mathrm{MDE_pBX}$[19], and FADE[20] are based on parameter selection, parameter adaption, strategy selection, and/or strategy adaption mechanisms. Several adaptive and self-adaptive mechanisms in DE are used for parameter selection/adaption and various strategies for pool selection/adaption. DE state-of-the-art proved to be very powerful by using conventional DE mutation variants to form a strategy pool in strategy selection/adaption or using parameter selection/adaption along with some conventional trial vector generation scheme. Some parameter and strategy adaption/selection schemes of DE state-of-the-art variant related work are discussed in this section.

Zaharie[13] introduced a parameter adaption scheme in his research work to control the diversity in the population. He used the concept of multipopulation for the diversity in the population that can help to avoid premature convergence. The experimental result showed promising performance for the proposed approach. Liu and Lampinen[20] introduced fuzzy-based control parameter adaption (FADE) in their research. They used a fuzzy controller to adopt the values of control parameters F and C_r in successive generations for the crossover and mutation operations of DE algorithm. Brest et al[2] introduced an adaptive control parameter setting of DE (jDE) by encoding F and C_r into individuals. They introduced τ_1 and τ_2 to control F and C_r parameter values in DE algorithm. The values of $\tau_1 = 0.1$, $\tau_2 = 0.1$, $F_u = 0.9$, $F_l = 0.1$ were used in their proposed adaption mechanism. In this technique, better individuals have opportunity to survive by propagating their better parameter values to produce new offspring. Qin et al[14] introduced both control parameter adaption as well as a strategy adaption mechanism in DE algorithm (SaDE). In the strategy adaption scheme, they used a strategy candidate pool of four strategies: `DE/rand/1`, `DE/rand/2`, `DE/rand to best/2`, and `DE/current to rand/1`. Each target vector generates a trial vector based on the learning period over previous generations that is based on the success rate of previous generations in the experiment. In parameter adaption, SaDE adjusts the control parameter C_r based on the median value of C_r that is calculated based on previous C_r values that have successfully generated a trial vector. Zhang and Sanderson[15] introduced a new parameter adaption method and used the `DE/current to pbest/1` strategy in their research work. In the parameter adaption, at each generation, C_r is generated by a normal distribution and F is generated by a Cauchy distribution to smoothly update these parameters. The `DE/current to pbest/1` strategy randomly selects the best population member from the top 100% population members of the current population. They used the concept that incorporation of a random best individual can enhance the searching capability of DE algorithm. JADE uses the concept of external archive to store the recently explored inferior solution and then select the new population members from the union of the current population and the external archive so that less performing individuals have a chance to be part of the population. Das et al[16] used `DE/target to best/1` by utilizing the concept of neighbourhood of each population individual in order to balance the exploration and exploitation abilities of DE algorithm. They introduced local- and global-neighbourhood DE (NSDE/DEGL) schemes and combined both local and global neighbourhood schemes to generate the donor vector. In NSDE, α and β are scaling factors, a local donor vector is created by employing the best vector in the neighbourhood and two other random vectors but the global best uses the entire population best vector and other two any random vectors from the current population. In DEGL, exploration and exploitation are controlled by a weight factor ω. They discussed increasing the weight factor, random weight factor, and self adaptive weight factor in their research work. Mallipeddi et al[18] introduced an ensemble-based crossover and mutation DE strategy (EPSDE) and their corresponding control parameter scheme in their research work. They used a pool of different crossover and mutation strategies and a pool of values for each associated control parameter. Each target vector generates a trial vector based on the assigned strategy and the parameter values. Successful combinations of the mutation and crossover strategy and associated parameter values are stored in the pool. EPSDE uses `DE/current to rand/1` and JADE mutation strategy along with binomial and exponential crossover. Minhazul Islam et al[19] proposed a new mutation strategy, a modification in the conventional binomial crossover scheme, and a parameter adaption scheme in DE algorithm. They used `DE/current to gr_best/1` and called it a less greedy version of `DE/current to best/1`. The proposed mutation strategy `DE/current to gr_best/1` uses the best vector from q% population of individuals to generate the trial vector for each target vector. In `pbest`, a crossover mutant vector can swap the p-top ranked individuals of the current generation instead of current parent using a binomial crossover scheme. In parameter adaption scheme, scale factor adaption is based on the Cauchy, and crossover probability adaption is based on the Gaussian distribution. Wang, Cai, and Zhang[17] introduced the composite DE (CoDE) variant. In this scheme, they used a pool of three trial vector generation strategies and a pool of three parameter setting combinations. The trial vector strategies used are `rand/1`, `rand/2`, and `current to rand/1`, while the parameter setting combinations are $[F = 1.0,\ C_r = 0.1]$, $[F = 1.0,\ C_r =$

0.9], and [$F = 0.8$, $C_r = 0.2$]. To generate a new solution in CoDE, each strategy is coupled with a randomly chosen parameter setting. Gong et al[21] introduced a new strategy adaptation mechanism (SaM) in their research work. They combined SaM with JADE and named it SaJADE. A novel strategy parameter $\eta_i \in [0,1)$ was used to control the selection of any strategy from the strategy pool. They chose four various DE strategies `DE/current to pbest` without archive, `DE/rand to pbest` without archive, `DE/current to pbest` with archive, and `DE/rand to pbest` with archive to form a strategy pool.

The popularity of DE due to its advantages over other evolutionary methods can be observed in its diverse applications in real life fields[22–29]. But the intensive and random development of DE algorithm has created several inconsistencies in the naming and formulation of trial vector generation schemes. The inconsistencies in DE algorithm may reduce the attraction of new researchers to use this algorithm in problem solving. It is important to mention that this study does not mean to prove any weakness in DE algorithm or criticize the effort of any researcher, but to make this algorithm easier and categorically clearer. Probably due to inconsistencies in DE mutation variants, there are some powerful conventional DE variants that are not announced to be powerful variants in the literature, while having either dominating or comparable performance as compared to those variants which are commonly used or used in DE state-of-the-art. Focusing on this, an effort is made in this study to present a consistent set of variants that DE has a number of variants and only few are commonly used that may also be due to inconsistency in most of DE variants. This study may not be a complete effort in this direction, but this attempt will prove to be significant addition in DE literature.

DE ALGORITHM

DE algorithm has three different parameters: a population of size N_{p}, a crossover control parameter C_r, and a difference vector amplification parameter F. Each population member in DE is represented as a D-dimensional parameter vector. In DE algorithm, the population is initialized randomly and is supposed to cover the entire search space. Each vector in the DE is represented by $\mathbf{x}_{i,g}$, where $i = 1,2,3,\dots,N_{\mathrm{p}}$ and g is generation number. New offsprings in DE algorithm are generated by mutation, crossover, and selection operators. The repair operator proposed by Wang[30] is also used in this study. Three different terminologies of vectors: donor vector, trial vector, and target vector are used in DE algorithm. Donor vector is a vector that is created in the mutation operation, trial vector is created in the crossover operation, and target vector is the current vector of population.

Mutation: In the mutation operation, a mutant vector, also called donor vector, is created. The donor vector $\mathbf{v}_{i,g}$ of the ith population member is calculated by adding the weighted difference of two vectors to the third vector:

$$\mathbf{v}_{i,g} = \mathbf{x}_{r_1,g} + F(\mathbf{x}_{r_2,g} - x_{\mathbf{r}_3,g}), \tag{1}$$

where indices $r_1, r_2, r_3 \in \{1,2,3,\dots,N_{\mathrm{p}}\}$ are randomly selected to be different from i and F is the mutation probability parameter.

Crossover: DE crossover strategies control the number of inherited components from the mutant vector to form a target vector. Binomial and exponential are main crossover schemes[16,31]. The DE crossover rate parameter C_r influences the size of the perturbation of the base (target) vector to ensure the population diversity[17].

Binomial crossover: In the crossover operation of DE algorithm, a trial vector is formed. In the binomial crossover scheme, the trail vector $\mathbf{u}_{i,g} = \langle u_{i,1,g}, u_{i,2,g}, \dots, u_{i,D,g} \rangle$ is generated by the equation, for $i = 1,2,\dots,N_{\mathrm{p}}$, $j = 1,2,\dots,D$:

$$u_{i,j,g} = \begin{cases} v_{i,j,g} & (\mathrm{rand}_j \leqslant C_r \text{ or } j = j_{\mathrm{rd}}), \\ x_{i,j,g}, & \text{otherwise}, \end{cases} \tag{2}$$

where j_{rd} is a randomly chosen integer in the range $[1, D]$, rand_j is a random number in $(0,1)$, $\mathbf{v}_{i,g}$ is the donor vector, and $C_r \in (0,1)$ is the crossover control parameter. Due to the range of j_{rand}, $\mathbf{u}_{i,g}$ is always different from $\mathbf{x}_{i,g}$.

Exponential crossover: In the exponential crossover scheme, the trail vector $\mathbf{u}_{i,g} = \langle u_{i,1,g}, u_{i,2,g}, \dots, u_{i,D,g} \rangle$ is created as:

$$u_{i,j,g} = \begin{cases} v_{i,j,g} & \text{if } j \in \{l, \langle l+1 \rangle_D, \dots, \langle l+L-1 \rangle_D\} \\ & \quad \text{and } (\mathrm{rand}_j \leqslant C_r), \\ x_{i,j,g} & \text{otherwise}, \end{cases} \tag{3}$$

for $i = 1,2,\dots,N_{\mathrm{p}}$, $j = 1,2,3,\dots,D$, where $\langle\cdot\rangle_D$ denotes the modulo function with modulus D. The starting index l is chosen at random from $[1, D]$ and L is also a randomly generated number from $[1, D]$. The parameters l and L are regenerated for each trial vector $\mathbf{u}_{i,g}$.

Selection: In DE algorithm, new population members are formed using the selection operation.

The selection operator uses the greedy approach by comparing the fitness of trial vector $\mathbf{u}_{i,g}$ with the fitness of target $\mathbf{x}_{i,g}$; the vector having best fitness is selected as a member of the new population:

$$\mathbf{x}_{i+1,g} = \begin{cases} \mathbf{u}_{i,g} & \text{fitness}(\mathbf{u}_{i,g}) < \text{fitness}(\mathbf{x}_{i,g}), \\ \mathbf{x}_{i,g}, & \text{otherwise}, \end{cases} \qquad (4)$$

where the fitness function calculates the fitness value of the objective function.

DE MUTATION VARIANTS, LITERATURE INCONSISTENCY AND SUGGESTED CORRECTIONS

There are several DE algorithm mutation variants/strategies that are formed by the linear combination of existing population members. The trial vector and target vector form the mutant vector in DE. Throughout this paper, x_i denotes the current target vector, i is the running index, u_i represents the trial vector, and v_i is a mutant vector. In DE algorithm, different mutation schemes are used to create the trial vector by using any combination of current, best, and random vectors. The behaviour of DE algorithm is influenced by the selection of mutation strategy and crossover scheme along with their control parameters: mutation probability F and crossover rate[31,32] C_r. The difference vector is the difference of two mutating vectors that is used to form offspring in the population[33]. To form the mutant vector in DE, some researchers use a random value $\lambda \in (0,1)$ as a coefficient multiplier with the first difference vector[1,34] and mutation probability F as a coefficient multiplier with the other difference vectors[34,35]. Some researchers have used only F as a coefficient multiplier with the difference vectors to form the mutant vector[31,36]. To reduce the number of control parameters of DE algorithm, we use[37] $\lambda = F$. This section contains the detail of DE mutation variants that reveals several irregularities in naming and formulation of DE mutation strategies. Before describing in detail the DE mutation variants, it is important to understand the vectors associated with DE mutation strategies. DE mutation strategies can be formed by the combinations of current, random, better, and best vectors. In any mutation strategy, the order, number, and name of vectors are very important. Throughout the analysis, x_g^k denotes the kth random vector for gth generation, v_g^i is the ith component of donor vector at gth generation, x_g^{best} states the best vector at gth generation, x_g^i denotes the current vector, also called target vector, at gth generation, and x_g^{better} states the better vector at gth generation.

The mutation strategies discussed below contain several inconsistencies that are identified in the next sections. The trial vector generation of each variant is presented by using selected population members that are based on the equations of associated variants. The important aspect is the order of vectors in the equation of variant and total number of vectors used to generate the trial vector generation for any variant.

(v_1) `DE/rand/1` was introduced by Storn & Price[1]. This strategy is known as the basic strategy in the DE algorithm. `DE/rand/1` is used as a default variant in the standard DE algorithm. This mutation strategy has no conflicts in the literature with other mutation strategies. The equation is

$$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^{r_3}), \qquad (5)$$

which uses three random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$ and one difference vector to generate the mutant vector. This strategy perturbs a based random vector $x_g^{r_1}$ with one weighted difference vector of random vectors.

(v_2) `DE/best/1` was introduced by Storn[34] for function optimization application. This mutation strategy has no conflicts with other mutation strategies in the literature. The equation is

$$v_g^i = x_g^{\text{best}} + F(x_g^{r_2} - x_g^{r_3}), \qquad (6)$$

which contains two random vectors $x_g^{r_1}$, $x_g^{r_2}$ in the difference vector and one best vector x_g^{best} to generate the mutant vector. This variant perturbs the best vector x_g^{best} with a weighted difference vector of random vectors.

(v_3) `DE/rand/2` was introduced by Storn & Price[31] for function optimization application. This mutation strategy has no conflicts with other mutation strategies in the literature. The equation is

$$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5}), \qquad (7)$$

which uses five random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$, $x_g^{r_5}$ in perturbation of a based random vector $x_g^{r_1}$ using two weighted difference vectors to generate the mutant vector. This results in a new vector that is not biased towards a particular direction.

(v_4) `DE/best/2` mutation strategy was introduced by Price[37] for function optimization problems.

This mutation strategy has no conflicts with other mutation strategies in the literature. The equation is

$$v_g^i = x_g^{\text{best}} + F(x_g^{r_1} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4}), \quad (8)$$

which perturbs x_g^{best} using four random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$ in two weighted difference vectors. This variant consumes best vector along with two weighted difference vectors without repeating any vector.

(v_5) `DE/current to rand/1` mutation strategy was used by many researchers[38–40]. This mutation strategy has naming conflict with other mutation strategies in the literature. The equation is

$$v_g^i = x_g^i + F(x_g^{r_1} - x_g^i) + F(x_g^{r_2} - x_g^{r_3}), \quad (9)$$

which places the perturbation at a location between the current population member x_g^i and a randomly chosen population member $x_g^{r_1}$ and uses a weighted difference vector of random vectors. This scheme uses three random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$ and the current vector x_g^i to generate the mutant vector.

(v_6) `DE/current to rand/1` mutation strategy was used by Qin et al[14] for constrained real parameter optimization. This mutation strategy has a naming conflict with other mutation strategies. The equation is

$$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^i) + F(x_g^{r_1} - x_g^{r_3}), \quad (10)$$

which places the perturbation at a location away from a random vector $x_g^{r_1}$ between the current vector and another random vector. This scheme contains three random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$ and the current vector x_g^i in the perturbation by repeating a random number $x_g^{r_1}$ in the weighted difference vector.

(v_7) `DE/current to best/1` mutation strategy, introduced by Storn & Price[1], was used by many researchers[41,42]. This mutation strategy has a naming and equation conflict with other mutation strategies in literature. The equation is

$$v_g^i = x_g^i + F(x_g^{\text{best}} - x_g^i) + F(x_g^{r_1} - x_g^{r_2}), \quad (11)$$

which places the perturbation at a location between the current vector x_g^i and the best vector x_g^{best} at current generation. The variant use a weighted difference of random vectors $x_g^{r_1}$, $x_g^{r_2}$ in perturbing the current vector x_g^i.

(v_8) `DE/current to best/1` mutation strategy was used in Podoba et al[43] for surface reconstruction using AI. This mutation strategy has a naming conflict as well as the equation conflict with other mutation strategies in the literature. The equation is

$$v_g^i = x_g^i + F(x_g^{\text{best}} - x_g^{r_1}) + F(x_g^{r_1} - x_g^{r_2}), \quad (12)$$

which perturbs the current vector x_g^i by moving the perturbation at a location between the best vector x_g^{best} and a random vector $x_g^{r_1}$. The variant utilizes two random vectors $x_g^{r_1}$, $x_g^{r_2}$, the best vector x_g^{best}, and the current vector x_g^i in perturbation with one weighted difference vector by repeating a random vector $x_g^{r_1}$.

(v_9) `DE/rand to best/1` mutation strategy was used in Davendra et al[44] for travelling salesman problem. This mutation strategy has a naming and equation conflict with the other mutation strategies in the literature. The equation is

$$v_g^i = x_g^i + F(x_g^{\text{best}} - x_g^{r_1}) + F(x_g^{r_1} - x_g^{r_2}), \quad (13)$$

which perturbs the current vector x_g^i by moving the at perturbation location between the best vector x_g^{best} and a random vector $x_g^{r_1}$. This perturbation uses two random vectors $x_g^{r_1}$, $x_g^{r_2}$, the best vector x_g^{best}, and the current vector x_g^i with one weighted difference vector by repeating a random vector $x_g^{r_1}$.

(v_{10}) `DE/rand to best/1` mutation strategy was introduced by Storn[34] for function optimization application. This mutation strategy has been used by many researchers[18,45]. This mutation strategy has naming conflict as well as equation conflict with other mutation strategies in the literature. The equation is

$$v_g^i = x_g^i + F(x_g^{\text{best}} - x_g^i) + F(x_g^{r_1} - x_g^{r_2}), \quad (14)$$

which perturbs the current vector x_g^i by moving perturbation at location between the current vector x_g^i and the best vector x_g^{best}. This perturbation use two random vectors $x_g^{r_1}$, $x_g^{r_2}$, the best vector x_g^{best}, and the current vector x_g^i with one weighted difference vector.

(v_{11}) `DE/rand to best/1` mutation strategy was used by many researchers[46]. This mutation strategy has naming conflict with other mutation strategies in literature. The equation is

$$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4}), \quad (15)$$

which perturbs a random vector $x_g^{r_1}$ by moving the perturbation at location between the best vector x_g^{best} and a random vector $x_g^{r_2}$. This perturbation uses four random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$ and the best vector x_g^{best} with one weighted difference vector.

(v_{12}) `DE/rand to best/1` mutation strategy was used by Almeida-Luz et al[47], Mendes & Mohais[48]. This mutation strategy has naming conflict with other mutation strategies in the literature. The equation is

$$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_1}) + F(x_g^{r_2} - x_g^{r_3}), \quad (16)$$

which perturbs a random vector $x_g^{r_1}$ by moving the perturbation at location between $x_g^{r_1}$ and the best vector x_g^{best}. This perturbation contains three random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$ and the best vector x_g^{best} with a weighted difference vector.

(v_{13}) `DE/rand to best/1` mutation strategy was used by Jeyakumar & Velayutham[40] for empirical analysis of DE variants over function optimization problem. This mutation strategy has naming conflict with other mutation strategies in the literature. The equation is

$$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^i) + F(x_g^{r_2} - x_g^{r_3}), \quad (17)$$

which perturbs a random vector $x_g^{r_1}$ by moving the perturbation at location between the best vector x_g^{best} and the current vector x_g^i. This perturbation contains three random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, the current vector x_g^i and the best vector x_g^{best} with one weighted difference vector.

(v_{14}) `DE/current to best/2` mutation strategy was used by many researchers[49,50] for various problems. This mutation strategy has naming and equation conflicts with other mutation strategies in the literature. The equation is

$$v_g^i = x_g^i + F(x_g^{\text{best}} - x_g^i) + F(x_g^{r_1} - x_g^{r_2}), \quad (18)$$

which perturbs the current vector x_g^i by moving the perturbation at location between the best vector x_g^{best} and the current vector x_g^i. This perturbation contains two random vectors $x_g^{r_1}$, $x_g^{r_2}$, the best vector x_g^{best} and the current vector x_g^i with one weighted difference vector.

(v_{15}) `DE/current to best/2` mutation strategy was used by many researchers[14,38,51]. This mutation strategy has naming conflict with other mutation strategies in literature. The equation of is

$$v_g^i = x_g^i + F(x_g^{\text{best}} - x_g^i) + F(x_g^{r_1} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4}), \quad (19)$$

which perturbs the current vector x_g^i by moving the perturbation at location between the best vector x_g^{best} and the current vector x_g^i. This perturbation contains four random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, the best vector x_g^{best} and the current vector x_g^i with two weighted difference vectors.

(v_{16}) `DE/current to rand/2` mutation strategy was used by Zielinski et al[38] for choosing suitable variants of differential evolution in particle swarm optimization. This mutation strategy has naming conflict with other mutation strategies in the literature. The equation is

$$v_g^i = x_g^i + F(x_g^{r_1} - x_g^i) + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5}), \quad (20)$$

which perturbs the current vector x_g^i by moving the perturbation at location between a random vector $x_g^{r_1}$ and the current vector x_g^i. This perturbation contains five random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$, $x_g^{r_5}$ and the current vector x_g^i with two weighted difference vectors.

(v_{17}) `DE/rand to best/2` mutation strategy was used by Podoba et al[43] for surface construction using AI. This mutation strategy has naming conflict with other mutation strategies in the literature. The equation is

$$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4}), \quad (21)$$

which perturbs a random vector $x_g^{r_1}$ by moving its perturbation location towards location between the best vector x_g^{best} and a random vector $x_g^{r_2}$. This perturbation contains four random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$ and the best vector x_g^{best} with one weighted difference vector.

(v_{18}) `DE/rand to best/2` mutation strategy was used by Oliveira & Saramago[52]. This mutation strategy has naming and equation conflicts with the other mutation strategies in the literature. The equation is

$$v_g^i = x_g^i + F(x_g^{\text{best}} - x_g^i) + F(x_g^{r_1} - x_g^{r_2}), \quad (22)$$

which perturbs the current vector x_g^i by moving the perturbation at location between the best vector x_g^{best} and the current vector x_g^i. This perturbation contains two random vectors $x_g^{r_1}$, $x_g^{r_2}$, the best vector x_g^{best} and the current vector x_g^i with one weighted difference vector.

(v_{19}) `DE/rand to best/2` mutation strategy was used by Weber[53] on parallel global optimization. This mutation strategy has naming conflict with other mutation strategies in the literature. The equation is

$$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^i) + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5}), \quad (23)$$

which perturbs a random vector $x_g^{r_1}$ by moving its perturbation location towards location between the best vector x_g^{best} and the current vector x_g^i. This perturbation contains five random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$, $x_g^{r_5}$, the current vector x_g^i and the best vector x_g^{best} with two weighted difference vectors.

(v_{20}) `DE/rand to best/2` mutation strategy was used by many researchers[45,54,55]. This mutation strategy has naming conflict with other mutation strategies in literature. The equation is

$$v_g^i = x_g^i + F(x_g^{\text{best}} - x_g^i) + F(x_g^{r_1} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4}), \quad (24)$$

which perturbs the current vector x_g^i by moving the perturbation at location between the best vector x_g^{best} and the current vector x_g^i. This perturbation contains four random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$, the current vector x_g^i and the best vector x_g^{best} with two weighted difference vectors.

(v_{21}) `DE/rand to best/2` mutation strategy was used by many researchers[39,56,57]. This mutation strategy has naming conflict with other mutation strategies in literature. The equation is

$$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_1}) + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5}), \quad (25)$$

which perturbs a random vector $x_g^{r_1}$ by moving the perturbation at location between the best vector x_g^{best} and a random vector $x_g^{r_1}$. This perturbation contains five random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$, $x_g^{r_5}$ and the best vector x_g^{best} with two weighted difference vectors.

(v_{22}) `DE/rand to current/2` mutation strategy was used by Elsayed et al[58,59]. This mutation strategy has no conflict with the other mutation strategies in the literature. The equation is

$$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^i) + F(x_g^{r_3} - x_g^{r_4}), \quad (26)$$

which perturbs a random vector $x_g^{r_1}$ by moving its perturbation location towards location between a random vector $x_g^{r_2}$ and the current vector x_g^i. This perturbation contains four random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$ and the current vector x_g^i with one difference vector.

(v_{23}) `DE/rand to best&current/2` mutation strategy was used by Elsayed et al[58,59]. This mutation strategy has no conflict with other mutation strategies in the literature. The equation is

$$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_2}) + F(x_g^{r_3} - x_g^i), \quad (27)$$

which perturbs a random vector $x_g^{r_1}$ by moving its perturbation location towards location between the best vector x_g^{best} and random vector $x_g^{r_2}$. This perturbation contains three random vectors $x_g^{r_1}$, $x_g^{r_2}$,$x_g^{r_3}$, the current vector x_g^i and a best vector x_g^{best} with a difference vector that utilizes the current vector.

(v_{24}) `DE/mid to better/1` mutation strategy was used by Xin et al[36] for designing powerful optimizer. This mutation strategy has no conflict with the other mutation strategies in the literature. The equation is

$$v_g^i = \tfrac{1}{2}F(x_g^{\text{bettle}} + x_g^i) + F(x_g^{\text{bettle}} - x_g^i) + F(x_g^{r_1} - x_g^{r_2}), \quad (28)$$

which perturbs the average of a better vector x_g^{better} and the current vector x_g^i by moving its perturbation location towards location between a better vector x_g^{better} and the current vector x_g^i. This perturbation contains two random vectors $x_g^{r_1}$, $x_g^{r_2}$, the current vector x_g^i and a better vector x_g^{better} with one weighted difference vector. This variant is different from other variants in the sense that it only uses a better vector along with average of a better vector and the current vector.

(v_{25}) `DE/rand/3` mutation strategy was used by Elsayed et al[58,59]. This mutation strategy has no conflict with the other mutation strategies in literature. The equation is

$$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5}) + F(x_g^{r_6} - x_g^{r_7}), \quad (29)$$

which contains seven random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$, $x_g^{r_5}$, $x_g^{r_6}$, $x_g^{r_7}$ in perturbation of a random vector $x_g^{r_1}$ with three weighted difference vectors.

(v_{26}) `DE/best/3` mutation strategy was used by Elsayed et al[58,59]. This mutation strategy has no conflict with the other mutation strategies in

Table 1 List of DE mutation variants available in the literature.

No.	Variant name	Equations
v_1	DE/rand/1	$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^{r_3})$
v_2	DE/best/1	$v_g^i = x_g^{\text{best}} + F(x_g^{r_2} - x_g^{r_3})$
v_3	DE/rand/2	$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5})$
v_4	DE/best/2	$v_g^i = x_g^{\text{best}} + F(x_g^{r_1} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4})$
v_5	DE/current to rand/1	$v_g^i = x_g^{i} + F(x_g^{r_1} - x_g^{i}) + F(x_g^{r_2} - x_g^{r_3})$
v_6	DE/current to rand/1	$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^{i}) + F(x_g^{r_1} - x_g^{r_3})$
v_7	DE/current to best/1	$v_g^i = x_g^{i} + F(x_g^{\text{best}} - x_g^{i}) + F(x_g^{r_1} - x_g^{r_2})$
v_8	DE/current to best/1	$v_g^i = x_g^{i} + F(x_g^{\text{best}} - x_g^{r_1}) + F(x_g^{r_1} - x_g^{r_2})$
v_9	DE/rand to best/1	$v_g^i = x_g^{i} + F(x_g^{\text{best}} - x_g^{r_1}) + F(x_g^{r_1} - x_g^{r_2})$
v_{10}	DE/rand to best/1	$v_g^i = x_g^{i} + F(x_g^{\text{best}} - x_g^{i}) + F(x_g^{r_1} - x_g^{r_2})$
v_{11}	DE/rand to best/1	$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4})$
v_{12}	DE/rand to best/1	$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_1}) + F(x_g^{r_2} - x_g^{r_3})$
v_{13}	DE/rand to best/1	$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{i}) + F(x_g^{r_2} - x_g^{r_3})$
v_{14}	DE/current to best/2	$v_g^i = x_g^{i} + F(x_g^{\text{best}} - x_g^{i}) + F(x_g^{r_1} - x_g^{r_2})$
v_{15}	DE/current to best/2	$v_g^i = x_g^{i} + F(x_g^{\text{best}} - x_g^{i}) + F(x_g^{r_1} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4})$
v_{16}	DE/current to rand/2	$v_g^i = x_g^{i} + F(x_g^{r_1} - x_g^{i}) + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5})$
v_{17}	DE/rand to best/2	$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4})$
v_{18}	DE/rand to best/2	$v_g^i = x_g^{i} + F(x_g^{\text{best}} - x_g^{i}) + F(x_g^{r_1} - x_g^{r_2})$
v_{19}	DE/rand to best/2	$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{i}) + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5})$
v_{20}	DE/rand to best/2	$v_g^i = x_g^{i} + F(x_g^{\text{best}} - x_g^{i}) + F(x_g^{r_1} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4})$
v_{21}	DE/rand to best/2	$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_1}) + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5})$
v_{22}	DE/rand to current/2	$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^{i}) + F(x_g^{r_3} - x_g^{r_4})$
v_{23}	DE/rand to best&current/2	$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{i})$
v_{24}	DE/mid to better/1	$v_g^i = \frac{1}{2}F(x_g^{\text{bettle}} + x_g^{i}) + F(x_g^{\text{bettle}} - x_g^{i}) + F(x_g^{r_1} - x_g^{r_2})$
v_{25}	DE/rand/3	$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5}) + F(x_g^{r_6} - x_g^{r_7})$
v_{26}	DE/best/3	$v_g^i = x_g^{\text{best}} + F(x_g^{r_1} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4}) + F(x_g^{r_5} - x_g^{r_6})$

literature. The equation is

$$v_g^i = x_g^{\text{best}} + F(x_g^{r_1} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4}) + F(x_g^{r_5} - x_g^{r_6}), \quad (30)$$

which contains the best vector x_g^{best} and six random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$, $x_g^{r_5}$, $x_g^{r_6}$ in perturbation of the best vector x_g^{best} with three weighted difference vectors.

DE is an emerging evolutionary algorithm; it contains a number of mutation variants/strategies. Various mutation strategies in the literature restrain irregularities with respect to naming of the variants and mathematical equation of variants. DE algorithm variants should be misinterpretation-free for its prosperity as an algorithm; to make this algorithm problem-free, an effort is made in this study to identify and remove the inconsistencies associated with DE algorithm variants in the literature. The detailed variants in terms of binomial and exponential schemes are reported in Table 1.

Inconsistencies in DE mutation strategies are identified with respect to their names and equations. Mathematical equations of variants have a key role since the implementation of each variant is carried out according to its mathematical equations.

IDENTIFICATION OF VARIANTS MATHEMATICAL EQUATION INCONSISTENCIES

Numerous variants in the literature contain the same mathematical equations but different names, which creates inconsistencies and lead to false impression. In this section, variants of Table 1 having the same mathematical equations and different names are identified.

Variants v_8 and v_9 have the same mathematical equation but different names, `DE/current to best/1` and `DE/rand to best/1`, respectively.

Variants v_7, v_{10}, v_{14}, and v_{18} have the same mathematical equations but different names, `DE/current to best/1`, `DE/rand to best/1`, `DE/current to best/2`, and `DE/rand to best/2`, respectively.

Variants v_{11} and v_{17} have the same mathematical equations but different names, `DE/rand to best/1` and `DE/rand to best/2`, respectively.

Variants v_{15} and v_{20} have the same mathematical equations but different names, `DE/current to best/2` and `DE/rand to best/2`, respectively.

IDENTIFICATION OF VARIANTS NAMING INCONSISTENCIES

There are many variants in the literature having the same name but different mathematical equations that fabricate misunderstanding for researchers.

Two variants v_5 and v_6 have the same name `DE/current to rand/1` but different mathematical equations.

Two variants v_7 and v_8 have the same name `DE/current to best/1` but different equations.

Five variants v_9–v_{13} have the same name `DE/rand to best/1` but different equations.

Two variants v_{14} and v_{15} have the same name `DE/current to best/2` but different equations.

Five variants v_{17}–v_{21} have the same name `DE/rand to best/2` but different equations.

THE PROPOSED SCHEME TO REMOVE NAMING AND EQUATION INCONSISTENCIES

The variant having the same mathematical equations but different names are combined because variant having the same equation produced the same representation and the same results that create problem for the users due to names. Variants having the same equations and different names are combined and reported in Table 2; variants v_8 and v_9 are combined as V_8 because they have the same mathematical equation, variants v_7, v_{10}, v_{14} and v_{18} are combined as V_7, variants v_{11} and v_{17} are combined as V_9, and variants v_{15} and v_{20} are combined as V_{12}. After combing these variants, they have assigned unique names refereeing the equation of that variant.

The naming inconsistency is removed by suggesting the consistent names and reported as suggested names in Table 2. The suggested names are relative to the equation of variant by considering number of vectors, the based vector and the arrangement of vectors used to form the equation of variant.

The variant V_5 has the same name as of variant V_6, `DE/current to rand/1`. V_5 has equation (9) that places the perturbation at a location between the current population member and a randomly chosen population member. One weighted difference vector is used in the equation of the variant. The equation of V_5 confirms the name of the variant `DE/current to rand/1`. However, V_6 has equation (10) that places the perturbation at a location away from a random vector $x_g^{r_1}$ between the current vector and another random vector. This scheme contains three random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$ and the current vector x_g^i in the perturbation by repeating a random number $x_g^{r_1}$ in the weighted difference vector. The suitable name of this variant relative to its equation is `DE/rand repeat&current to rand/1`, because a based random vector $x_g^{r_1}$ is repeated in the weighted difference vector and the perturbation location is between the current vector and a random vector.

The variants V_7–V_{11} have naming inconsistency with each other. Variant V_7 is used in the literature with various names, `DE/current to best/1`, `DE/rand to best/1`, `DE/current to best/2`, and `DE/rand to best/2`. V_7 has equation (11) that places the perturbation at a location between the current vector x_g^i and the best vector x_g^{best} at the current generation. The variant uses one weighted difference vector of random vectors $x_g^{r_1}$, $x_g^{r_2}$ in perturbing the current vector x_g^i. The suitable name for this variant relative to its equation is `DE/current to best/1`.

Variant V_8 has names `DE/current to best/1` and `DE/rand to best/1` in the literature and has equation (12), which perturbs the current vector x_g^i by moving perturbation at a location between the best vector x_g^{best} and a random vector $x_g^{r_1}$. The variant uses two random vectors $x_g^{r_1}$, $x_g^{r_2}$, the best vector x_g^{best} and the current vector x_g^i in perturbation with one weighted difference vector by repeating a random vector $x_g^{r_1}$. The suitable name of this variant relative to its equation is `DE/current&rand repeat to best/1`.

Variant V_9 has names `DE/rand to best/1` and `DE/rand to best/2` in the literature and has equation (15), which perturbs a random vector $x_g^{r_1}$ by moving perturbation at a location between best vector x_g^{best} and a random vector $x_g^{r_1}$. This perturbation uses four random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$ and the best vector with one weighted difference

Table 2 List of DE variants after removing equation and name inconsistencies.

No.	Variant name(s)	Equations	Suggested name†
V_1	`DE/rand/1`	$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^{r_3})$	
V_2	`DE/best/1`	$v_g^i = x_g^{\text{best}} + F(x_g^{r_2} - x_g^{r_3})$	
V_3	`DE/rand/2`	$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5})$	
V_4	`DE/best/2`	$v_g^i = x_g^{\text{best}} + F(x_g^{r_1} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4})$	
V_5	`DE/current to rand/1`	$v_g^i = x_g^i + F(x_g^{r_1} - x_g^i) + F(x_g^{r_2} - x_g^{r_3})$	`DE/current to rand/1`
V_6	`DE/current to rand/1`	$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^i) + F(x_g^{r_1} - x_g^{r_3})$	`DE/rand repeat&current to rand/1`

V_7	`DE/current to best/1`
	`DE/rand to best/1`
	`DE/current to best/2`
	`DE/rand to best/2`
		$v_g^i = x_g^i + F(x_g^{\text{best}} - x_g^i) + F(x_g^{r_1} - x_g^{r_2})$	`DE/current to best/1`

V_8	`DE/current to best/1` `DE/rand to best/1`	$v_g^i = x_g^i + F(x_g^{\text{best}} - x_g^{r_1}) + F(x_g^{r_1} - x_g^{r_2})$	`DE/current&rand repeat to best/1`
V_9	`DE/rand to best/1` `DE/rand to best/2`	$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4})$	`DE/rand to best/1`
V_{10}	`DE/rand to best/1`	$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_1}) + F(x_g^{r_2} - x_g^{r_3})$	`DE/rand repeat to best/1`
V_{11}	`DE/rand to best/1`	$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^i) + F(x_g^{r_2} - x_g^{r_3})$	`DE/rand&current to best/1`
V_{12}	`DE/current to best/2` `DE/rand to best/2`	$v_g^i = x_g^i + F(x_g^{\text{best}} - x_g^i) + F(x_g^{r_1} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4})$	`DE/current to best/2`
V_{13}	`DE/current to rand/2`	$v_g^i = x_g^i + F(x_g^{r_1} - x_g^i) + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5})$	
V_{14}	`DE/rand to best/2`	$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^i) + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5})$	`DE/rand&current to best/2`
V_{15}	`DE/rand to best/2`	$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_1}) + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5})$	`DE/rand repeat to best/2`
V_{16}	`DE/rand to current/2`	$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^i) + F(x_g^{r_3} - x_g^{r_4})$	`DE/rand&current to rand/1`
V_{17}	`DE/rand to best&current/2`	$v_g^i = x_g^{r_1} + F(x_g^{\text{best}} - x_g^{r_2}) + F(x_g^{r_3} - x_g^i)$	`DE/rand to best&current/1`
V_{18}	`DE/mid to better/1`	$v_g^i = \frac{1}{2}F(x_g^{\text{bettle}} + x_g^i) + F(x_g^{\text{bettle}} - x_g^i) + F(x_g^{r_1} - x_g^{r_2})$	
V_{19}	`DE/rand/3`	$v_g^i = x_g^{r_1} + F(x_g^{r_2} - x_g^{r_3}) + F(x_g^{r_4} - x_g^{r_5}) + F(x_g^{r_6} - x_g^{r_7})$	
V_{20}	`DE/best/3`	$v_g^i = x_g^{\text{best}} + F(x_g^{r_1} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4}) + F(x_g^{r_5} - x_g^{r_6})$	

† Variants with name inconsistencies are suggested with consistent name.

vector. The suitable name of this variant relative to its equation is `DE/rand to best/1`.

Both variants V_{10} and V_{11} have the name `DE/rand to best/1` in literature, which is the same as V_9. V_{10} has equation (16) that perturbs a random vector $x_g^{r_1}$ by moving perturbation at a location between $x_g^{r_1}$ and the best vector x_g^{best}. This perturbation contains three random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$ and the best vector x_g^{best} with one weighted difference vector. The suggested name of this variant relative to its equation is `DE/rand repeat to best/1`.

V_{11} has equation (17) that perturbs a random vector $x_g^{r_1}$ by moving perturbation at a location between the best vector x_g^{best} and the current vector x_g^i. This perturbation contains three random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, the current vector $x_g^{r_1}$ and the best vector x_g^{best} with one weighted difference vector. The suggested name of this variant relative to its equation is `DE/rand&current to best/1`.

The variants V_{12}, V_{14}, V_{15} have naming inconsistency with one another. V_{12} in the literature has names `DE/current to best/2` and `DE/rand to best/2` and has equation (19), which perturbs the current vector x_g^i by moving perturbation at a location between the best vector x_g^{best} and the current vector. This perturbation contains four random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$, the best vector x_g^{best} and the current vector x_g^i with two weighted difference vectors. The suggested name of this variant relative to its equation is `DE/current to best/2`.

In the literature, both variants V_{14} and V_{15} have the name `DE/rand to best/2`, which is also used for V_{12}. V_{14} has equation (23) that perturbs a random vector $x_g^{r_1}$ by moving perturbation at a location between the best vector x_g^{best} and the current vector x_g^i. This perturbation contains five random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$, $x_g^{r_5}$, the current vector x_g^i and the best vector x_g^{best} with two weighted difference vectors. The suggested name of this variant relative to its equation is `DE/rand&current to best/2`.

V_{15} has equation (25) that perturbs a random vector $x_g^{r_1}$ by moving perturbation at location between the best vector x_g^{best} and a random vector $x_g^{r_1}$. This perturbation contains five random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$, $x_g^{r_5}$ and the best vector x_g^{best} with two weighted difference vectors. The suitable name of this variant relative to its equation is `DE/rand repeat to best/2`.

The variant V_{16} with name `DE/rand to current/2` in the literature has equation (26) that perturbs a random vector $x_g^{r_1}$ by moving perturbation at a location between a random vector $x_g^{r_2}$ and the current vector x_g^i. This perturbation contains four random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, $x_g^{r_4}$ and the current vector x_g^i with one difference vector. The suggested name of this variant relative to its equation is `DE/rand&current to rand/1`.

The variant V_{17} with name `DE/rand to best&current/2` in literature has equation (27) that perturbs a random vector $x_g^{r_1}$ by moving perturbation at a location between the best vector x_g^{best} and a random vector $x_g^{r_2}$. This perturbation contains three random vectors $x_g^{r_1}$, $x_g^{r_2}$, $x_g^{r_3}$, the current vector x_g^i and the best vector x_g^{best} with one difference vector that utilizes the current vector x_g^i. The suggested name of this variant relative to its equation is `DE/rand to best&current/1`.

To summarize, mathematical equation inconsistencies in DE mutation variants are removed from Table 1 and reported in Table 2, which still contains naming inconsistencies. Inconsistent names are suggested with new names according to the corresponding equations and reported in Table 2, which are used in the paper to generate results. The short names of DE mutation variants that will be used throughout the remaining paper for binomial variants are V_1–V_{20} with corresponding V_{21}–V_{40} for exponential variants.

Table 3 Test suit of benchmark functions.

Function	Equation
f_1	$f(x)=\sum_{i=0}^{n} x_i^2$
f_2	$f(x)=\sum_{i=0}^{n}\left(\sum_{j=0}^{i} x_i^2\right)^2$
f_3	$f(x)=\sum_{i=1}^{n-1}\left[100(x_{i+1}-x_i^2)^2+(1-x_i^2)\right]$
f_4	$f(x)=\sum_{i=1}^{n}\left[\frac{1}{4000}x_i^2-\prod_{j=1}^{n}\cos\frac{x_j}{\sqrt{j}}+1\right]$
f_5	$f(x)=-20\,\mathrm{e}^{-0.2\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}}-\mathrm{e}^{\frac{\sum_{i=1}^{n}\cos 2\pi x_i}{n}}+20+\mathrm{e}$
f_6	$f(x)=\sum_{i=1}^{n}\lfloor x_i+0.5\rfloor^2$

EXPERIMENTAL RESULTS AND DISCUSSION

To illustrate the varying performance of DE mutation strategies, a test suit of benchmark n-dimensional functions taken from Ref. 2 is used (Table 3). The experimental results of average fitness values of DE mutation strategies are obtained over 30 independent runs, 2000 training iterations, 10 dimensions, population size $N_\text{p}=30$, crossover probability $C_r=0.5$, and mutation rate $F=0.7$ in the experimentation. The main aim of this section is to show the diverse performance of DE mutation strategies given in Table 2. The DE mutation strategies are used as V_1–V_{20} for binomial mutation and V_{21}–V_{40} for the corresponding exponential mutation.

The convergence graphs are generated showing iterations along the x-axis and the performance along the y-axis. The variants are from V_1–V_{20} for binomial variants and V_{21}–V_{40} for the exponential version of each corresponding binomial variant. The convergence graphs are also generated using the same experimental setting for each mutation strategy and function. Convergence graphs of two sample functions f_1 (Fig. 1) and f_3 (Fig. 2) are given to show the distinct performance of each variant. Convergence graphs show that each DE mutation strategy has its own optimization curve.

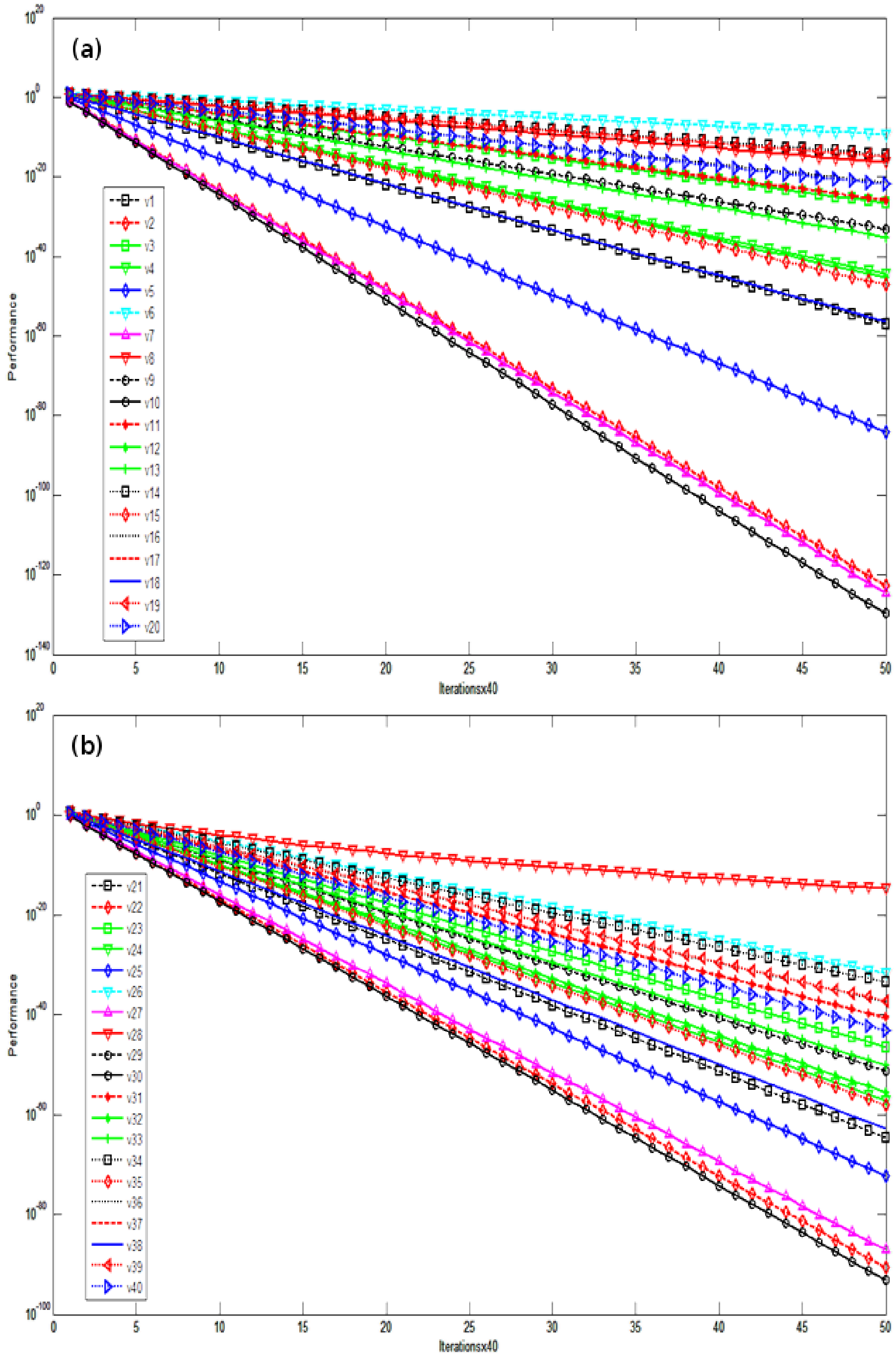

Fig. 1 Logarithmic convergence graphs of average fitness of functions f_1 for (a) V_1–V_{20} and (b) V_{21}–V_{40} showing the average fitness versus the number of iterations.

From experimental results, it is observed that all 40 mutation strategies of DE algorithm are valid strategies to enhance the functionality of DE algorithm, and hence can be effectively used wherever DE algorithm is used.

The average fitness values of DE mutation strategies for testing suit of functions are given in Table 4. The results are obtained using the same experimental setting for each variant and function. The average fitness results of DE mutation strategies

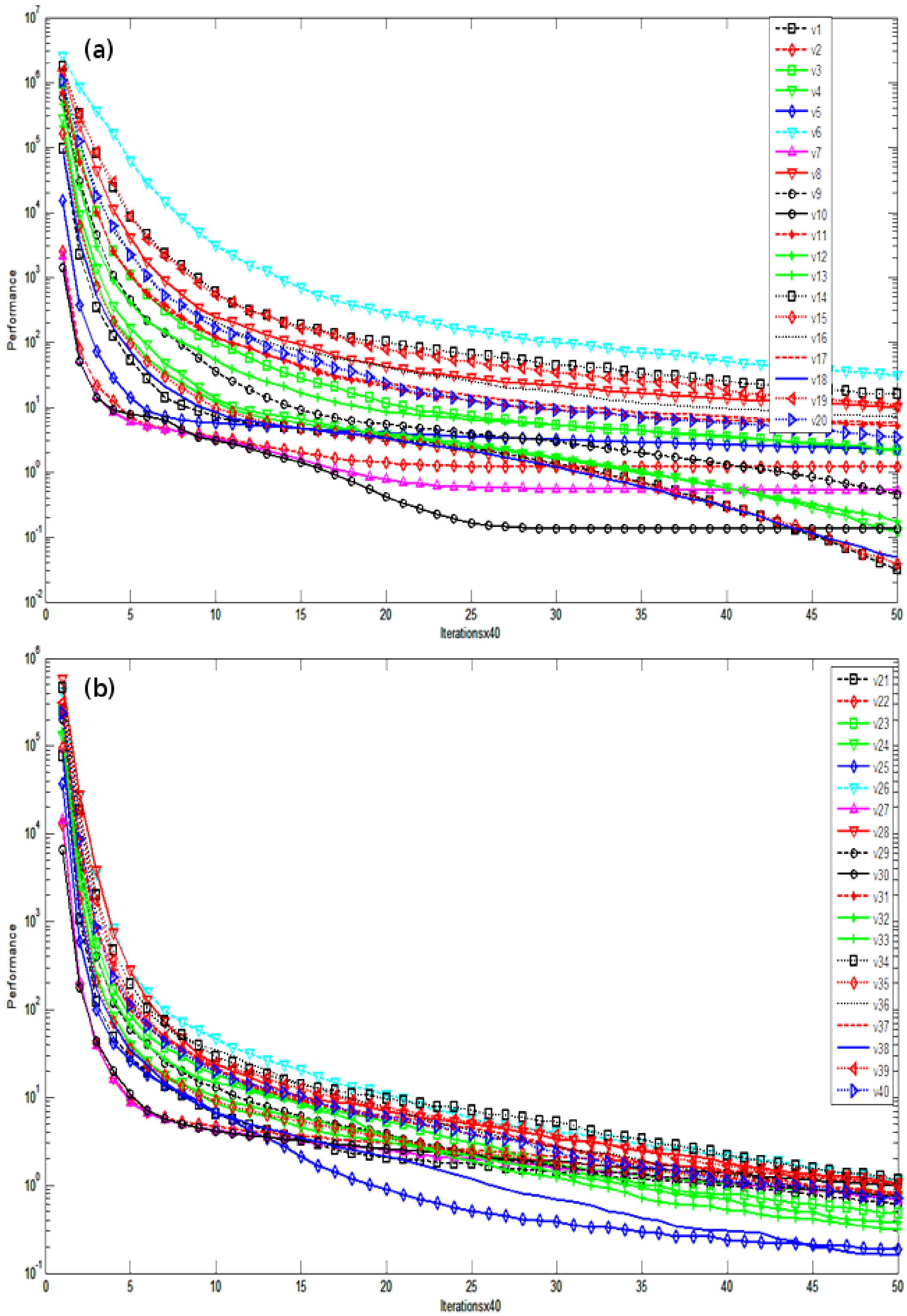

Fig. 2 Logarithmic convergence graphs of average fitness of functions f_3 for (a) V_1–V_{20} and (b) V_{21}–V_{40} showing the average fitness versus the number of iterations.

show distinct performance of each variant for each function.

To explore the diverse performance of DE variants, the ranks of DE variants are calculated. The ranks (Table 5) show that each DE mutation strategies have different performance from others. The rank of each variant is calculated based on the of average fitness values of that variant for each function. The values show the corresponding rank of each variant in ascending order; the smaller the rank, the

Table 4 Average fitness results of DE variants of functions f_1–f_6.

Var.	f_1	f_2	f_3	f_4	f_5	f_6
V_1	$(1.3\pm2.9)\times10^{-57}$	$(4.2\pm4.6)\times10^{-10}$	$(3.1\pm1.4)\times10^{-2}$	$(2.7\pm4.1)\times10^{-2}$	$(1.0\pm1.4)\times10^{-1}$	$(1.5\pm1.3)\times10^{-11}$
V_2	$(1.8\pm8.5)\times10^{-123}$	$(1.4\pm2.3)\times10^{-32}$	1.2 ± 1.8	$(5.4\pm3.1)\times10^{-2}$	$(2.7\pm0.7)\times10^{-1}$	$(0.6\pm1.7)\times10^{-11}$
V_3	$(2.6\pm3.3)\times10^{-27}$	$(7.0\pm5.9)\times10^{-3}$	2.1 ± 0.6	$(2.7\pm0.5)\times10^{-1}$	$(5.9\pm0.6)\times10^{-1}$	$(2.1\pm2.0)\times10^{-11}$
V_4	$(0.4\pm1.2)\times10^{-44}$	$(3.7\pm5.3)\times10^{-7}$	$(1.1\pm0.8)\times10^{-1}$	$(2.2\pm0.6)\times10^{-1}$	$(4.3\pm1.2)\times10^{-1}$	$(2.7\pm2.5)\times10^{-11}$
V_5	$(0.9\pm1.3)\times10^{-84}$	$(1.1\pm1.0)\times10^{-19}$	2.2 ± 1.3	$(4.5\pm3.6)\times10^{-2}$	$(3.2\pm0.5)\times10^{-1}$	$(2.4\pm2.4)\times10^{-11}$
V_6	$(4.4\pm4.7)\times10^{-10}$	$(1.9\pm1.3)\times10^{1}$	$(3.0\pm1.4)\times10^{1}$	$(3.7\pm0.6)\times10^{-1}$	$(8.5\pm0.6)\times10^{-1}$	$(5.6\pm5.4)\times10^{-11}$
V_7	$(2.7\pm9.3)\times10^{-125}$	$(0.4\pm1.0)\times10^{-35}$	0.5 ± 1.3	$(2.8\pm1.9)\times10^{-2}$	$(2.3\pm0.8)\times10^{-1}$	$(1.3\pm1.9)\times10^{-11}$
V_8	$(4.2\pm3.0)\times10^{-17}$	$(5.8\pm3.3)\times10^{-3}$	9.9 ± 2.7	$(3.8\pm0.7)\times10^{-1}$	$(8.0\pm0.8)\times10^{-1}$	$(4.2\pm4.2)\times10^{-11}$
V_9	$(7.8\pm8.5)\times10^{-34}$	$(1.9\pm1.5)\times10^{-4}$	$(4.5\pm1.7)\times10^{-1}$	$(2.5\pm0.4)\times10^{-1}$	$(5.3\pm0.6)\times10^{-1}$	$(1.8\pm1.9)\times10^{-11}$
V_{10}	$(2.2\pm7.1)\times10^{-130}$	$(2.1\pm9.9)\times10^{-32}$	$(1.3\pm7.1)\times10^{-1}$	$(2.1\pm1.5)\times10^{-2}$	$(8.6\pm7.8)\times10^{-2}$	$(1.4\pm1.8)\times10^{-11}$
V_{11}	$(1.3\pm1.5)\times10^{-26}$	$(9.0\pm8.5)\times10^{-2}$	5.1 ± 0.8	$(3.0\pm0.5)\times10^{-1}$	$(6.1\pm0.9)\times10^{-1}$	$(3.1\pm2.9)\times10^{-11}$
V_{12}	$(4.7\pm6.9)\times10^{-46}$	$(4.0\pm5.7)\times10^{-8}$	$(1.7\pm2.7)\times10^{-1}$	$(2.4\pm0.5)\times10^{-1}$	$(5.1\pm0.6)\times10^{-1}$	$(3.9\pm3.9)\times10^{-11}$
V_{13}	$(6.8\pm1.5)\times10^{-36}$	$(2.5\pm2.1)\times10^{-5}$	2.2 ± 1.5	$(2.5\pm0.5)\times10^{-1}$	$(5.2\pm0.7)\times10^{-1}$	$(2.0\pm1.9)\times10^{-11}$
V_{14}	$(4.1\pm4.8)\times10^{-15}$	4.5 ± 1.8	$(1.5\pm0.4)\times10^{1}$	$(3.6\pm0.7)\times10^{-1}$	$(7.7\pm0.7)\times10^{-1}$	$(3.1\pm3.0)\times10^{-11}$
V_{15}	$(7.9\pm11)\times10^{-48}$	$(1.3\pm1.5)\times10^{-7}$	$(3.7\pm2.9)\times10^{-2}$	$(2.2\pm0.5)\times10^{-1}$	$(4.7\pm0.5)\times10^{-1}$	$(2.7\pm4.0)\times10^{-11}$
V_{16}	$(4.2\pm3.7)\times10^{-22}$	$(5.2\pm3.8)\times10^{-1}$	7.1 ± 1.3	$(3.1\pm0.5)\times10^{-1}$	$(6.6\pm0.7)\times10^{-1}$	$(2.6\pm2.7)\times10^{-11}$
V_{17}	$(1.8\pm4.1)\times10^{-26}$	$(8.2\pm5.3)\times10^{-2}$	5.5 ± 1.6	$(2.9\pm0.4)\times10^{-1}$	$(5.7\pm0.6)\times10^{-1}$	$(2.4\pm2.0)\times10^{-11}$
V_{18}	$(4.0\pm6.3)\times10^{-57}$	$(4.5\pm5.4)\times10^{-10}$	$(4.8\pm7.2)\times10^{-2}$	$(1.6\pm0.5)\times10^{-1}$	$(4.4\pm0.5)\times10^{-1}$	$(2.3\pm2.3)\times10^{-11}$
V_{19}	$(9.9\pm7.9)\times10^{-16}$	1.2 ± 0.8	10.0 ± 7.9	$(3.6\pm0.4)\times10^{-1}$	$(7.1\pm0.7)\times10^{-1}$	$(2.8\pm3.6)\times10^{-11}$
V_{20}	$(1.1\pm1.4)\times10^{-22}$	$(6.4\pm5.3)\times10^{-2}$	3.0 ± 1.0	$(3.4\pm0.5)\times10^{-1}$	$(6.6\pm0.8)\times10^{-1}$	$(2.4\pm2.4)\times10^{-11}$
V_{21}	$(0.5\pm1.5)\times10^{-64}$	$(1.0\pm1.1)\times10^{-8}$	$(5.1\pm6.5)\times10^{-1}$	$(0.5\pm1.8)\times10^{-3}$	$(1.2\pm3.7)\times10^{-2}$	$(3.6\pm3.6)\times10^{-11}$
V_{22}	$(5.0\pm8.2)\times10^{-92}$	$(3.0\pm3.8)\times10^{-17}$	$(3.2\pm9.8)\times10^{-1}$	$(2.1\pm2.0)\times10^{-2}$	$(9.0\pm8.3)\times10^{-2}$	$(1.1\pm2.0)\times10^{-11}$
V_{23}	$(3.7\pm6.4)\times10^{-47}$	$(1.6\pm1.1)\times10^{-4}$	$(6.7\pm5.8)\times10^{-1}$	$(0.3\pm1.3)\times10^{-3}$	$(0.4\pm2.2)\times10^{-2}$	$(6.3\pm4.8)\times10^{-11}$
V_{24}	$(4.0\pm4.5)\times10^{-58}$	$(3.2\pm5.1)\times10^{-7}$	$(7.8\pm7.4)\times10^{-1}$	$(8.0\pm8.5)\times10^{-3}$	$(2.2\pm5.1)\times10^{-2}$	$(5.5\pm4.3)\times10^{-11}$
V_{25}	$(0.7\pm1.3)\times10^{-72}$	$(7.9\pm7.8)\times10^{-13}$	$(3.2\pm6.4)\times10^{-1}$	$(3.9\pm4.7)\times10^{-3}$	$(1.8\pm0.4)\times10^{-1}$	$(3.5\pm3.4)\times10^{-11}$
V_{26}	$(2.1\pm1.6)\times10^{-32}$	$(2.8\pm2.1)\times10^{-1}$	1.1 ± 0.7	$(5.8\pm5.3)\times10^{-3}$	$(0.4\pm2.1)\times10^{-3}$	$(7.4\pm7.3)\times10^{-11}$
V_{27}	$(0.7\pm1.0)\times10^{-87}$	$(0.6\pm1.0)\times10^{-18}$	$(6.9\pm9.4)\times10^{-1}$	$(6.5\pm5.5)\times10^{-3}$	$(1.7\pm0.3)\times10^{-1}$	$(5.0\pm6.3)\times10^{-11}$
V_{28}	$(1.9\pm2.2)\times10^{-15}$	$(1.9\pm1.6)\times10^{-4}$	$(9.9\pm8.7)\times10^{-1}$	$(2.5\pm1.0)\times10^{-2}$	$(2.0\pm0.3)\times10^{-1}$	$(1.2\pm1.0)\times10^{-10}$
V_{29}	$(6.2\pm7.8)\times10^{-52}$	$(1.0\pm1.0)\times10^{-5}$	$(7.2\pm6.1)\times10^{-1}$	$(1.0\pm3.3)\times10^{-6}$	$(0.4\pm2.5)\times10^{-2}$	$(5.7\pm6.8)\times10^{-11}$
V_{30}	$(1.2\pm1.9)\times10^{-93}$	$(1.8\pm1.8)\times10^{-17}$	0.9 ± 1.2	$(0.8\pm2.4)\times10^{-3}$	$(0.4\pm2.2)\times10^{-2}$	$(3.5\pm3.5)\times10^{-11}$
V_{31}	$(2.1\pm1.8)\times10^{-41}$	$(9.9\pm9.6)\times10^{-3}$	0.9 ± 1.1	$(2.3\pm3.8)\times10^{-3}$	$(7.2\pm7.4)\times10^{-12}$	$(6.2\pm6.3)\times10^{-11}$
V_{32}	$(1.6\pm2.3)\times10^{-56}$	$(0.7\pm1.1)\times10^{-7}$	$(5.9\pm7.0)\times10^{-1}$	$(6.8\pm5.7)\times10^{-3}$	$(0.2\pm2.7)\times10^{-2}$	$(3.3\pm3.0)\times10^{-11}$
V_{33}	$(1.0\pm1.7)\times10^{-50}$	$(2.9\pm1.9)\times10^{-6}$	$(4.0\pm2.5)\times10^{-1}$	$(6.3\pm6.1)\times10^{-3}$	$(0.2\pm5.4)\times10^{-2}$	$(5.7\pm6.2)\times10^{-11}$
V_{34}	$(3.3\pm3.4)\times10^{-34}$	$(9.5\pm4.8)\times10^{-2}$	$(8.3\pm5.1)\times10^{-1}$	$(6.4\pm5.3)\times10^{-3}$	$(1.7\pm1.5)\times10^{-11}$	$(5.0\pm5.0)\times10^{-11}$
V_{35}	$(8.1\pm9.5)\times10^{-59}$	$(2.0\pm1.8)\times10^{-7}$	$(6.1\pm6.2)\times10^{-1}$	$(0.3\pm1.7)\times10^{-3}$	$(2.2\pm2.1)\times10^{-12}$	$(5.6\pm5.6)\times10^{-11}$
V_{36}	$(2.1\pm1.7)\times10^{-38}$	$(2.9\pm1.5)\times10^{-2}$	$(7.5\pm5.3)\times10^{-1}$	$(1.9\pm3.0)\times10^{-3}$	$(8.8\pm9.6)\times10^{-12}$	$(0.6\pm1.0)\times10^{-10}$
V_{37}	$(3.2\pm4.9)\times10^{-41}$	$(9.9\pm8.2)\times10^{-3}$	$(6.9\pm6.0)\times10^{-1}$	$(1.6\pm3.8)\times10^{-3}$	$(7.0\pm5.5)\times10^{-12}$	$(5.8\pm5.5)\times10^{-11}$
V_{38}	$(1.6\pm3.6)\times10^{-63}$	$(1.6\pm1.6)\times10^{-8}$	$(1.7\pm3.1)\times10^{-1}$	$(7.8\pm6.5)\times10^{-3}$	$(7.5\pm6.6)\times10^{-2}$	$(3.4\pm3.4)\times10^{-11}$
V_{39}	$(6.1\pm8.0)\times10^{-38}$	$(1.2\pm0.8)\times10^{-2}$	1.2 ± 0.8	$(3.5\pm5.0)\times10^{-3}$	$(4.3\pm4.0)\times10^{-12}$	$(5.0\pm5.1)\times10^{-11}$
V_{40}	$(0.9\pm1.1)\times10^{-43}$	$(4.8\pm5.4)\times10^{-4}$	$(8.6\pm5.8)\times10^{-1}$	$(4.0\pm5.3)\times10^{-3}$	$(1.6\pm4.2)\times10^{-2}$	$(7.5\pm5.6)\times10^{-11}$

better performance of the mutation strategy.

CONCLUSIONS

The selection of a DE variant affects the performance result of DE algorithm since there is a deviation in performance of DE variants. DE algorithm variants in the literature have naming and mathematical equation inconsistencies. This study identifies these inconsistencies and presents a consistent set of DE variants. The details of naming and formulation inconsistencies is discussed. The naming and formulation inconsistencies are removed based on the number of vectors and order of vectors used to form the equation of variant. This study will prove to be a significant addition in DE literature in view of the fact that the existence of any inconsistency might trigger off new researchers. Further studies are still required to explore the limitations, advantages, and flaws in this direction. The main focus in this work is not the overwhelming existing DE variants but the new tracks to work on the DE variants in optimization. Future work of this study is to perform a thorough analysis of these DE variants to access the performance of each DE

Table 5 Average fitness ranks of DE variants of functions.

Var.	f_1	f_2	f_3	f_4	f_5	f_6
V_1	13	9	1	22	17	5
V_2	3	2	28	25	24	1
V_3	31	27	30	32	33	8
V_4	21	18	4	27	26	15
V_5	7	4	32	24	25	11
V_6	40	40	40	39	40	30
V_7	2	1	13	23	23	3
V_8	36	26	37	40	39	25
V_9	29	24	11	31	31	6
V_{10}	1	3	5	19	15	4
V_{11}	32	34	34	34	34	18
V_{12}	20	13	6	29	29	24
V_{13}	27	21	31	30	30	7
V_{14}	39	39	39	38	38	17
V_{15}	18	15	2	28	28	14
V_{16}	35	37	36	35	35	13
V_{17}	33	33	35	33	32	12
V_{18}	14	10	3	26	27	9
V_{19}	37	38	38	37	37	16
V_{20}	34	32	33	36	36	10
V_{21}	9	11	12	4	11	23
V_{22}	5	7	8	20	16	2
V_{23}	19	22	16	2	8	37
V_{24}	12	17	21	18	13	29
V_{25}	8	8	9	10	21	22
V_{26}	30	36	27	12	7	38
V_{27}	6	5	18	15	18	27
V_{28}	38	23	26	21	22	40
V_{29}	16	20	19	1	10	33
V_{30}	4	6	24	5	8	21
V_{31}	23	28	25	8	4	36
V_{32}	15	14	14	16	20	19
V_{33}	17	19	10	13	19	32
V_{34}	28	35	22	14	6	26
V_{35}	11	16	15	3	1	31
V_{36}	25	31	20	7	5	35
V_{37}	24	29	17	6	3	34
V_{38}	10	12	7	17	14	20
V_{39}	26	30	29	9	2	28
V_{40}	22	25	23	11	12	39

variant that may prove to be a significant addition in DE literature. Another possible direction of future work is to develop more powerful parent selection schemes that may improve the performance of these variants and to develop a mathematical model for DE variants that can serve as a standard.

REFERENCES

1. Storn R, Price K (1995) *Differential Evolution: a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*, ICSI, Berkeley, CA.
2. Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans Evol Comput* **10**, 646–57.
3. Price KV, Storn RM, Lampinen JA (2005) *Differential Evolution: a Practical Approach to Global Optimization*, Springer-Verlag, Berlin, Heidelberg.
4. Xia S, Zhou M, Li G (2011) A binary adaptive differential evolution algorithm for dynamic economic dispatch considering significant wind power. In: *PowerTech, 2011 IEEE Trondheim*, IEEE, pp 1–8.
5. Yang S, Qing A (2005) Design of high-power millimeter-wave TM/sub 01/-TE/sub 11/mode converters by the differential evolution algorithm. *IEEE Trans Plasma Sci* **33**, 1372–6.
6. Zhao SZ, Qu BY, Suganthan PN, Iruthayarajan MW, Baskar S (2010) Multi-objective robust PID controller tuning using multi-objective differential evolution. In: *11th International Conference on Control Automation Robotics & Vision*, IEEE, pp 2398–403.
7. Chiou JP, Wang FS (1998) A hybrid method of differential evolution with application to optimal control problems of a bioprocess system. In: *The 1998 IEEE International Conference on Evolutionary Computation Proceedings*, IEEE, pp 627–32.
8. Fang Q, Chen D, Yu H, Wu X (2004) Differential evolution algorithm based on eugenic strategy and its application to chemical engineering. *J Chem Ind Eng* **55**, 598–602. [in Chinese]
9. Das S, Abraham A, Konar A (2008) Automatic clustering using an improved differential evolution algorithm. *IEEE Trans Syst Man Cybern A* **38**, 218–37.
10. Du JX, Huang DS, Wang XF, Gu X (2007) Shape recognition based on neural networks trained by differential evolution algorithm. *Neurocomputing* **70**, 896–903.
11. Karaboga N (2005) Digital IIR filter design using differential evolution algorithm. *EURASIP J Adv Signal Process* **2005**, 856824.
12. Das S, Abraham A, Konar A (2008) Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives. In: Liu Y, Sun A, Loh HT, Lu WF, Lim EP (eds) *Advances of Computational Intelligence in Industrial Systems*, Springer, pp 1–38.
13. Zaharie D (2003) Control of population diversity and adaptation in differential evolution algorithms. In: Matousek R, Osmera P (eds) *Proceedings of Mendel 9th International Conference on Soft Computing*, pp 41–6.
14. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans Evol Comput* **13**, 398–417.
15. Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. *IEEE*

Trans Evol Comput **13**, 945–58.
16. Das S, Abraham A, Chakraborty UK, Konar A (2009) Differential evolution using a neighborhood-based mutation operator. *IEEE Trans Evol Comput* **13**, 526–53.
17. Wang Y, Cai Z, Zhang Q (2011) Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans Evol Comput* **15**, 55–66.
18. Mallipeddi R, Suganthan PN, Pan QK, Tasgetiren MF (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl Soft Comput* **11**, 1679–96.
19. Islam SM, Das S, Ghosh S, Roy S, Suganthan PN (2012) An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. *IEEE Trans Syst Man Cybern B* **42**, 482–500.
20. Liu J, Lampinen J (2005) A fuzzy adaptive differential evolution algorithm. *Soft Comput* **9**, 448–62.
21. Gong W, Cai Z, Ling CX, Li C (2011) Enhanced differential evolution with adaptive strategies for numerical optimization. *IEEE Trans Syst Man Cybern B* **41**, 397–413.
22. Basu M (2008) Optimal power flow with FACTS devices using differential evolution. *Int J Electr Power Energ Syst* **30**, 150–6.
23. Rocca P, Oliveri G, Massa A (2011) Differential evolution as applied to electromagnetics. *IEEE Antenn Propag Mag* **53**, 38–49.
24. Qin H, Zhou J, Lu Y, Li Y, Zhang Y (2010) Multi-objective cultured differential evolution for generating optimal trade-offs in reservoir flood control operation. *Water Resour Manag* **24**, 2611–32.
25. Suresh K, Kundu D, Ghosh S, Das S, Abraham A, Han SY (2009) Multi-objective differential evolution for automatic clustering with application to micro-array data analysis. *Sensors* **9**, 3981–4004.
26. Selbaş R, Kızılkan O, Reppich M (2006) A new design approach for shell-and-tube heat exchangers using genetic algorithms from economic point of view. *Chem Eng Process* **45**, 268–75.
27. Aslantas V, Kurban R (2010) Fusion of multi-focus images using differential evolution algorithm. *Expert Syst Appl* **37**, 8861–70.
28. Ilonen J, Kamarainen JK, Lampinen J (2003) Differential evolution training algorithmfor feed-forward neural networks. *Neural Process Lett* **17**, 93–105.
29. Vasundhara Mandal D, Ghoshal SP, Kar R (2013) Digital FIR filter design using hybrid random particle swarm optimization with differential evolution. *Int J Comput Intell Syst* **6**, 911–27.
30. Storn R, Price KV (1996) Minimizing the real functions of the ICEC'96 contest by differential evolution. In: *Proceedings of IEEE International Conference on Evolutionary Computation*, pp 842–4.
31. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* **11**, 341–59.
32. Ting CK, Huang CH (2009) Varying number of difference vectors in differential evolution. In: *IEEE Congress on Evolutionary Computation*, IEEE, pp 1351–8.
33. Sai Hanuman A, Hyd G (2010) Data clustering using almost parameter free differential evolution technique. *Int J Comput Appl* **8**, 1–7.
34. Storn R (1996) On the usage of differential evolution for function optimization. In: *Biennial Conference of the North American Fuzzy Information Processing Society*, IEEE, pp 519–23.
35. Brest J, Bošković B, Greiner S, Žumer V, Maučec MS (2007) Performance comparison of self-adaptive and adaptive differential evolution algorithms. *Soft Comput* **11**, 617–29.
36. Xin B, Chen J, Zhang J, Fang H, Peng ZH (2012) Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy. *IEEE Trans Syst Man Cybern C* **42**, 744–67.
37. Price KV (1996) Differential evolution: a fast and simple numerical optimizer. In: *Biennial Conference of the North American Fuzzy Information Processing Society*, IEEE, pp 524–7.
38. Zielinski K, Joost M, Laur R, Orlik B (2008) Choosing suitable variants of differential evolution and particle swarm optimization for the optimization of a PI cascade control. In: *11th International Conference on Optimization of Electrical and Electronic Equipment*, IEEE, pp 55–62.
39. Gong W, Cai Z (2012) Adaptive parameter selection for strategy adaptation in differential evolution for continuous optimization. *J Comput* **7**, 672–9.
40. Jeyakumar G, Shunmuga Velayutham C (2010) An empirical performance analysis of differential evolution variants on unconstrained global optimization problems. *Int J Comput Inform Syst Ind Manag Appl* **2**, 77–86.
41. García-Martínez C, Rodríguez FJ, Lozano M (2011) Role differentiation and malleable mating for differential evolution: an analysis on large-scale optimisation. *Soft Comput* **15**, 2109–26.
42. Gong W, Cai Z, Jia L, Li H (2011) A generalized hybrid generation scheme of differential evolution for global numerical optimization. *Int J Comput Intell Appl* **10**, 35–65.
43. Podoba T, Tomsu L, Vlcek K, Heczko M (2010) Surface reconstruction by means of AI. In: *International Conference on Computational & Experimental Engineering and Sciences*, vol 15, 111–22.
44. Zelinka I, Senkerik R, Bialic-Davendra M, Davendra D (2010) Chaos driven evolutionary algorithm for the traveling salesman problem. In: Davendra D (ed) *Traveling Salesman Problem, Theory and Applications*, InTech, pp 55–70.

45. Triguero I, García S, Herrera F (2011) Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification. *Pattern Recogn* **44**, 901–16.
46. Zhao SZ, Suganthan PN, Das S (2011) Self-adaptive differential evolution with multi-trajectory search for large-scale optimization. *Soft Comput* **15**, 2175–85.
47. Almeida-Luz S, Vega-Rodríguez MA, Gómez-Pulido JA, Sánchez-Pérez JM (2008) Applying differential evolution to the reporting cells problem. In: *Proceedings of the International Multiconference on Computer Science and Information Technology*, IEEE, pp 65–71.
48. Mendes R, Mohais AS (2005) DynDE: a differential evolution for dynamic optimization problems. In: *The 2005 IEEE Congress on Evolutionary Computation*, IEEE, pp 2808–15.
49. Ao YY, Chi HQ (2010) An adaptive differential evolution algorithm to solve constrained optimization problems in engineering design. *Engineering* **2**, 65–77.
50. Mandal A, Das S, Abraham A (2011) A differential evolution based memetic algorithm for workload optimization in power generation plants. In: *11th International Conference on Hybrid Intelligent Systems*, IEEE, pp 271–6.
51. Epitropakis MG, Tasoulis DK, Pavlidis NG, Plagianakos VP, Vrahatis MN (2011) Enhancing differential evolution utilizing proximity-based mutation operators. *IEEE Trans Evol Comput* **15**, 99–119.
52. de Oliveira GTS, Saramago SFP (2007) A contribution to the study about differential evolution. *Ciênc Eng* **16**, 1–8.
53. Weber M (2010) Parallel global optimization: structuring populations in differential evolution. PhD thesis, Univ of Jyvaskyla, Finland.
54. Triguero I, García S, Herrera F (2010) A preliminary study on the use of differential evolution for adjusting the position of examples in nearest neighbor classification. In: *IEEE Congress on Evolutionary Computation*, IEEE, pp 1–8.
55. Goudos SK, Zaharis ZD, Yioultsis TV (2010) Application of a differential evolution algorithm with strategy adaptation to the design of multi-band microwave filters for wireless communications. *Progr Electromagn Res* **109**, 123–37.
56. Fialho A, Schoenauer M, Sebag M (2010) Fitness-AUC bandit adaptive strategy selection vs the probability matching one within differential evolution: an empirical comparison on the bbob-2010 noiseless testbed. In: *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*, Oregon, USA, pp 1535–42.
57. Fialho A, Gong W, Cai Z (2010) Probability matching-based adaptive strategy selection vs uniform strategy selection within differential evolution: an empirical comparison on the bbob-2010 noiseless testbed. In: *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*, Oregon, USA, pp 1527–34.
58. Elsayed SM, Sarker RA, Essam DL (2011) Differential evolution with multiple strategies for solving CEC2011 real-world numerical optimization problems. In: *2011 IEEE Congress on Evolutionary Computation*, IEEE, pp 1041–8.
59. Elsayed SM, Sarker RA, Essam DL (2011) Multi-operator based evolutionary algorithms for solving constrained optimization problems. *Comput Oper Res* **38**, 1877–96.