

Received March 31, 2021, accepted April 29, 2021, date of publication May 4, 2021, date of current version May 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077242

# Differential Evolution Mutations: Taxonomy, Comparison and Convergence Analysis

ALI WAGDY MOHAMED<sup>1,2</sup>, ANAS A. HADI<sup>3</sup>, AND ALI KHATER MOHAMED<sup>4</sup>

<sup>1</sup>Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza 12613, Egypt

<sup>2</sup>Wireless Intelligent Networks Center (WINC), School of Engineering and Applied Sciences, Nile University, Giza 12588, Egypt

<sup>3</sup>College of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia

<sup>4</sup>Faculty of Computer Science, October University for Modern Sciences and Arts (MSA), Giza 12451, Egypt

Corresponding author: Ali Wagdy Mohamed (aliwagdy@gmail.com)

**ABSTRACT** During last two decades, Differential Evolution (DE) proved to be one of the most popular and successful evolutionary algorithms for solving global optimization problems over continuous space. Proposing new mutation strategies to improve the optimization performance of (DE) is considered a significant research study. In DE, mutation operation plays a vital role in the performance of the algorithm. Therefore, in this paper, comprehensive analysis of the contributions on basic and novel mutation strategies that were proposed between 1995 and 2020 is presented. A new taxonomy based on the structure of the novel mutations is proposed. Numerical experiments on a set of 30 test problems from the CEC2017 benchmark for 10, 30, 50 and 100 dimensions, including a comparison with classical DE schemes and recent mutations schemes are executed. Furthermore, theoretical, and empirical convergence behavior analysis of all mutations is discussed. The paper also presents many recommendations, guidelines, insights, and suggestions for experienced practitioners and interested researchers in designing and developing effective and efficient DE algorithms to address various optimization problems in different fields.

**INDEX TERMS** Evolutionary computation, global optimization, differential evolution, mutation strategy, taxonomy, correct and false convergence.

## I. INTRODUCTION

Differential Evolution (DE) proposed by Storn and Price [1], [2], is a stochastic population-based search method. It exhibits excellent capability in solving a wide range of optimization problems with different characteristics from several fields and many real-world application problems. Similar to all other Evolutionary algorithms (EAs), the evolutionary process of DE uses mutations, crossover, and selection operators at each generation to reach the global optimum. Besides, it is one of the most efficient evolutionary algorithms (EAs) currently in use. In DE, each individual in the population is called target vector. Mutation is used to generate a mutant vector, which perturbs a target vector using the difference vector of other individuals in the population. After that, crossover operation generates the trial vector by combining the parameters of the mutation vector with the parameters of a parent vector selected from the population. Finally, according to the fitness value and selection operation determines which of the vectors should be chosen for the

next generation by implementing a one-to-one completion between the generated trial vectors and the corresponding parent vectors [3], [4]. The performance of DE basically depends on the mutation strategy, the crossover operator. Besides, the intrinsic control parameters (population size NP, scaling factor F, the crossover rate CR) play a vital role in balancing the diversity of population and convergence speed of the algorithm [5]–[8]. The advantages are the simplicity of implementation, reliability, speed, and robustness [9], [10]. Thus, it has been widely applied in solving many real-world applications of science and engineering, such as {0-1} Knapsack Problem [11], [12], financial markets dynamic modeling [13], feature selection [14], admission capacity planning higher education [15], [16], and solar energy [17], for more applications, interested readers can refer to [18]. However, DE has many weaknesses, as all other evolutionary search techniques do w.r.t the “no free lunch” (NFL) theorem. Generally, DE has a good global exploration ability that can reach the region of global optimum, but it is slow at the exploitation of the solution [19]. Additionally, the parameters of DE are problem dependent and it is difficult to adjust them for different problems. Moreover, DE performance

The associate editor coordinating the review of this manuscript and approving it for publication was Sunith Bandaru<sup>1</sup>.

decreases as search space dimensionality increases [20]–[22]. Finally, the performance of DE deteriorates significantly when the problems of premature convergence and/or stagnation occur [20]. Consequently, researchers have suggested many techniques to improve the basic DE. From the literature [18], [23]–[28], these proposed modifications, improvements, and developments on DE focus on adjusting control parameters in an adaptive or self-adaptive manner while there are a few attempts in developing new mutations rule. In fact, the main objective of this study is to discuss the advantages and disadvantages of these mutations and provide future recommendations, guidelines, insights, and suggestions for experienced practitioners and interested researchers in designing and developing effective and efficient DE algorithms.

Therefore, to accomplish this objective, the main originality our work in this paper goes in four directions. The first direction is to provide a comprehensive review to the contributions on basic and novel mutation strategies that were proposed between 1995 and 2020 to investigate the similarities and differences in their mathematical and/or probabilistic structure and design.

Thus, for this purpose, which is the second direction of our work, two novel taxonomies are proposed in this research study. Based on the structure of the mutations, the first taxonomy (section III, Figure.2) is proposed to eliminate any ambiguity related to classifying any DE mutation and as a helper tool for all researchers to fully understand and differentiate between mutation strategies. The new theoretical classification comprises two research area regarding the contribution type to the mutation strategies of Differential Evolution algorithm (Novel Mutation and Novel concept). In the first area, an innovative mutation scheme is added to basic DE mutation. However, unlike novelty of mutation, regarding the second area, a new technique is proposed to enhance the selection process of the individuals to form the mutation scheme. Based on the structure of the novel mutations, it can be further classified into three groups: (a) random, (b) directed, and (c) probabilistic. Then, based on the design of mutation scheme, the directed mutation can be classified into two main categories: (1) partially directed and (2) fully directed.

On the other hand, based on experimental results, analysis and comparisons of the performance of all mutations on CEC2017 benchmark test problems [29] with 10, 30, 50 and 100 dimensions, the second taxonomy (section IV-Figure 4) is a new taxonomy proposed to classify the Mutation strategies of DE algorithms into four categories (algorithms show excellent performance and continuous improvement, algorithms show unstable performance and slight diminishes, algorithms show descent performance with complete and/or significant deterioration, algorithms show stable moderate or poor performance with insignificant improvement).

The third direction of this work is to statistically compare and analyze the performance of mutations. Besides, the convergence behavior of top ten algorithms and the superior

performance of all algorithms that provided minimum mean function error on the CEC 2017 functions with all dimensions are presented. Finally, detailed theoretical background, definitions, explanation, and new classifications for correct and false convergence scenarios are given. Besides, based on the results provided by all mutations, numerical experiments, and complete analysis of the convergence behavior for all mutations are presented.

To the best of our knowledge, this is the first study that reviews all these different types of mutations, proposes theoretical classifications for all mutations and carry out empirical evaluation and comparison.

The rest of the paper is organized as follows. Section II gives a brief introduction to canonical DE algorithm, including its typical mutation operators, crossover, and selection operators. Section III provides the first taxonomy that classifies and reviews the published work on mutations of DE algorithms. Next, in Section IV, the proposed second taxonomy based on experimental results and comparison of all DE mutations. Besides, statistical analysis, convergence analysis and superior analysis for all algorithms are introduced. Section V discusses the theoretical and empirical convergence behavior for all mutations. In section VI, future recommendations, guidelines, insights, and suggestions for experienced practitioners and interested researchers in designing and developing effective and efficient DE algorithms are presented. Finally, Section VII concludes the paper and summarizes the objectives addressed.

## II. BASIC DIFFERENTIAL EVOLUTION

This section provides a brief summary of the basic Differential Evolution algorithm. In a simple DE, generally known as DE/rand/1/bin [2], [30], an initial random population, denoted by  $P$ , consists of  $NP$  individual. Each individual is represented by the vector  $x_i = (x_{1,i}, x_{2,i}, \dots, x_{D,i})$ , where  $D$  is the number of dimensions in solution space. Since the population will be varied with the running of evolutionary process, the generation times in DE are expressed by  $G = 0, 1 \dots, Gmax$ , where  $Gmax$  is the maximal times of generations. For the  $i$ th individual of  $P$  at the  $G$  generation, it is denoted by  $x_i^G = (x_{1,i}^G, x_{2,i}^G, \dots, x_{D,i}^G)$ . The lower and upper bounds in each dimension of search space are respectively recorded by  $X_L = (x_{1,L}, x_{2,L}, \dots, x_{D,L})$  and  $X_U = (x_{1,U}, x_{2,U}, \dots, x_{D,U})$ . The initial population  $P_0$  is randomly generated according to a uniform distribution within the lower and upper boundaries ( $X_L, X_U$ ). After initialization, these individuals are evolved by DE operators (mutation and crossover) to generate a trial vector. A comparison between the parent and its trial vector is then done to select the vector which should survive to the next generation [23], [31]. DE steps are discussed below:

### A. INITIALIZATION

In order to establish a starting point for the optimization process, an initial population  $P_0$  must be created. Typically, each  $j^{\text{th}}$  component ( $j = 1, 2, \dots, D$ ) of the  $i$ th individuals

( $i = 1, 2, \dots, NP$ ) in the  $P_0$  is obtained as follow:

$$x_{j,i}^0 = x_{j,L} + rand(0, 1) \cdot (x_{j,U} - x_{j,L}) \quad (1)$$

where  $rand(0, 1)$  returns a uniformly distributed random number in  $[0, 1]$ .

### B. MUTATION

At generation  $G$ , for each target vector  $x_i^G$ , a mutant vector  $v_i^G$  is generated according to the following:

$$v_i^G = x_{r_1}^G + F \cdot (x_{r_2}^G - x_{r_3}^G), r_1 \neq r_2 \neq r_3 \neq i \quad (2)$$

where  $r_1, r_2, r_3 \in \{1, 2, \dots, NP\}$  are three randomly chosen indices.  $F$  is a real number to control the amplification of the difference vector  $(x_{r_2}^G - x_{r_3}^G)$ . According to Storn and Price [2], the range of  $F$  is in  $[0, 2]$ . In this work, if a component of a mutant vector violates search space, then the new value of this component is generated a new by (1).

### C. CROSSOVER

There are two main crossover types, binomial, and exponential. We here elaborate the binomial crossover. In the binomial crossover, the target vector is mixed with the mutated vector, using the following scheme, to yield the trial vector  $u_{j,i}^G$ .

$$u_{j,i}^G = \begin{cases} v_{j,i}^G, & \text{if } (rand \leq CR \text{ or } j = j_{rand}) \\ x_{j,i}^G, & \text{otherwise} \end{cases} \quad (3)$$

where  $rand_{j,i}(i \in [1, NP]$  and  $j \in [1, D])$  is a uniformly distributed random number in  $[0, 1]$ ,  $CR \in [0, 1]$  called the crossover rate that controls how many components are inherited from the mutant vector,  $j_{rand}$  is a uniformly distributed random integer in  $[1, D]$  that makes sure at least one component of trial vector is inherited from the mutant vector.

### D. SELECTION

DE adapts a greedy selection strategy. If and only if the trial vector  $U_i^G$  yields as good as or a better fitness function value than  $X_i^G$ , then  $U_i^G$  is set to  $X_i^{G+1}$ . Otherwise, the old vector  $X_i^G$  is retained. The selection scheme is as follows (for a minimization problem):

$$x_i^{G+1} = \begin{cases} u_i^G, & \text{if } f(u_i^G) \leq f(x_i^G) \\ x_i^G, & \text{otherwise} \end{cases} \quad (4)$$

A detailed description of standard DE algorithm is given in Figure 1.

## III. MUTATIONS IN DIFFERENTIAL EVOLUTION

DE/rand/1 is the fundamental mutation strategy developed by Storn and Price [1], [2] and is reported to be the most successful and widely used scheme in the literature [23]. Obviously, in this strategy, the three vectors are chosen from the population at random for mutation and the base vector is then selected at random among the three. The other two vectors form the difference vector that is added to the base vector. Consequently, it is able to maintain population diversity and global search capability with no bias to any specific

```

1  Generate the initial population
2  Evaluate the fitness of each individual
3  While the termination criteria is not satisfied do
4      For  $i=1$  to  $NP$  do
5          Select uniform randomly  $r_1 \neq r_2 \neq r_3 = i$ 
6               $j_{rand} = rndint(1, D)$ 
7          For  $j=1$  to  $D$  do
8              If  $rndreal_j(0,1) < CR$  or  $j$  is equal to  $j_{rand}$  then
9                   $u_{i,j} = x_{r_1,j} + F \cdot (x_{r_2,j} - x_{r_3,j})$ 
10             else
11                  $u_{i,j} = x_{i,j}$ 
12             Endif
13         End for
14     End for
15     For  $i=1$  to  $NP$  do
16         Evaluate the offspring  $u_i$ 
17         If  $f(u_i)$  is better than or equal to  $f(x_i)$  then
18             Replace  $x_i$  with  $u_i$ 
19         Endif
20     End for
21 End While

```

**FIGURE 1.** Description of standard DE algorithm.  $rand [0,1]$  is a function that returns a real number between 0 and 1.  $randint(\min, \max)$  is a function that returns an integer number between min and max.  $rndreal(0,1)$  returns a real random number between 0 and 1.  $NP$ ,  $G_{max}$ ,  $CR$  and  $F$  are user-defined parameters.  $D$  is the dimensionality of the problem.

search direction, but it slows down the convergence speed of DE algorithms [32]. Thus, proposing new mutation strategies to improve the optimization performance of differential evolution (DE) are considered an important research study. Virtually, during the past 15 years, many researchers have been working on the improvement of the mutation strategies of DE.

In fact, the contribution to the mutation strategies of Differential Evolution algorithm is divided into two main areas, depending on the contribution proposed. The two areas are (1) novel mutation and (2) novel concept.

In the first area, an innovative mutation scheme is added to basic DE mutation. However, unlike novelty of mutation, regarding the second area, a new technique or method is proposed to enhance the selection process of the individuals to form the mutation scheme. Thus, these methods have been applied using the existing mutation schemes without any modifications or improvement to the existing mutation schemes.

Based on the structure of the novel mutation, it can be further classified into three groups: (a) Random, (b) directed, and (c) probabilistic. Then, the directed mutation can be classified into two main categories: (1) partially directed and (2) fully directed.

Concerning the first group, random mutations, the mutant vector is generated or created using randomly selected individuals in the current population i.e., random mutation does not include the incorporation of the objective function value of the selected individuals in the mutation scheme itself. However, in contrast to random mutations, directed mutations includes the incorporation of the objective function value of the selected individuals in the mutation scheme itself.

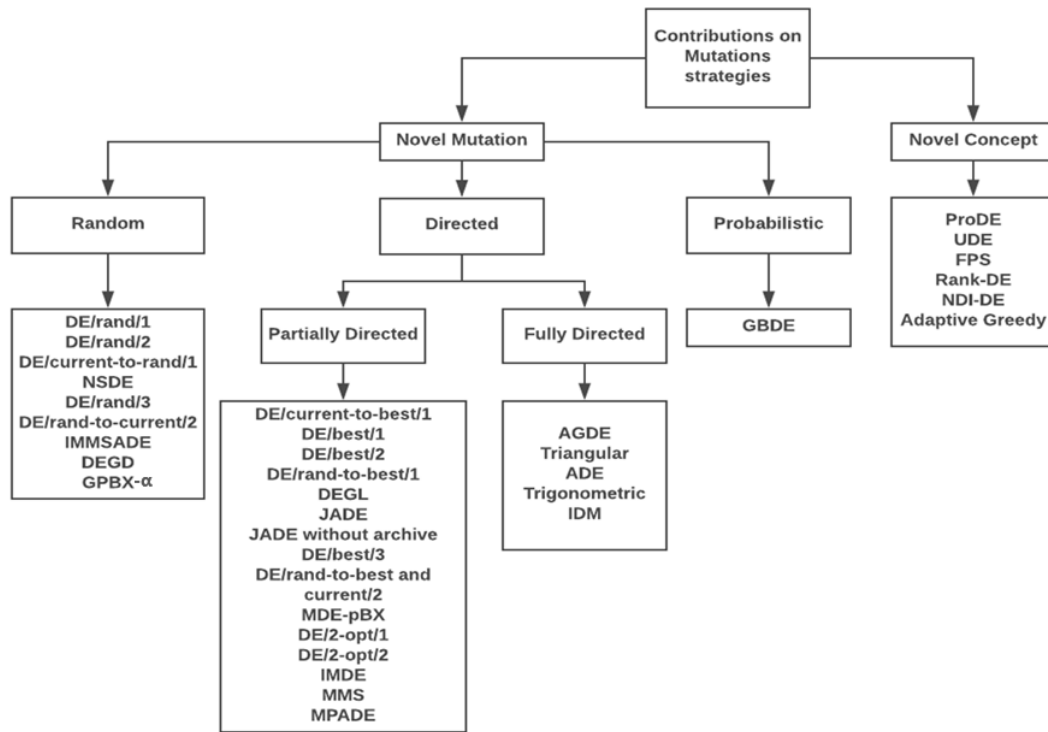


FIGURE 2. New classification illustrating the position of every mutation scheme according to the suggested taxonomy.

Therefore, the partially directed mutations, the mutant vector is generated or created using some randomly selected individuals in addition to the neighbor best or worst vector(s) and/or global best or worst vector(s) found so far in the current population, it must be noted that any difference vector in mutation scheme includes best or worst information, not both or the base vector is either best vector or worst vector. Besides, regarding full directed mutations category, the mutant vector is generated or created using some randomly selected individuals in addition to the neighbor best and worst vector(s) and/or global best and worst vector(s) found so far in the current population, thus, some difference vectors is directed from worst vector(s) to best vector(s) i.e., both vectors will be used to form the mutation scheme. Alternatively, it is based on randomly selected individuals that are sorted in ascending order, from best to worst, according to their objective function values. On the contrary, regarding probabilistic mutations group, the mutant vector is generated or created using a probabilistic distribution such as uniform, Gaussian, Cauchy, and other well-known distributions. Taken into consideration that the required parameters of the selected distribution are determined based on the available statistical information about the entire population such as neighbor/global best vector(s), mean, median, worst, and standard deviation vector(s). It must be noted that the effectiveness of all novel concept techniques has been previously proved as excellent alternative methods for improving the performance of basic and novel mutations. Virtually, the main objective of this study is to evaluate the individual effect

of each mutation in solving CEC 2017 test functions under the same experimental conditions. Thus, in order to exclude joint effect i.e. (novel or basic mutation combined with the novel concept), it is not considered for further evaluation and comparison. Actually, based on the structure of the mutations, it is noteworthy to mention that this is the first research paper that proposes a new taxonomy to classify the contributions of DE mutations. As listed in Figure 2 and Table 1, there are 30 novel mutations and 6 novel concepts that have been reviewed and classified according to the new taxonomy. They will be discussed in the next subsections.

### A. NOVEL MUTATIONS

In this subsection, we will briefly discuss the innovative mutation schemes that are added to basic DE mutation.

#### 1) RANDOM MUTATIONS

*a: DE/RAND/1*

Price and Storn [1] presented a new heuristic approach for solving non-continuous and non-differentiable continuous space functions called Differential Evolution. The authors tested many variants in order to represent the most promising variant DE/rand/1. In which, the mutant vector is generated for each individual in the population by randomly selecting 3 mutually exclusive vectors that are different from the base vector using the following equation:

$$v_{i,g} = x_{r1,g} + F * (x_{r2,g} - x_{r3,g})$$

$$r1, r2, r3 \in [1, NP], r1, \neq r2 \neq r3 \neq i \quad (5)$$

TABLE 1. The classification of mutations schemes.

Classification	Type	Mutation	Year	Ref.	
Novel Mutation	Random	DE/rand/1	1995	[1]	
		DE/rand/2	2005	[33]	
		DE/current-to-rand/1	2006	[34]	
		NSDE	2007	[35]	
		DE/rand/3	2009	[37]	
		DE/rand-to-current/2	2011	[36]	
		DEGD	2011	[38]	
		GPBX- $\alpha$	2011	[39]	
		IMMSADE	2017	[40]	
	Directed	Partially directed	DE/current-to-best/1	1995	[1]
			DE/best/1	2005	[33]
			DE/best/2	2005	[33]
			DE/rand-to-best and current/2	2006	[41]
			DE/rand-to-best/1	2009	[42]
			DEGL	2009	[20]
			JADE	2009	[43]
			JADE without archive	2009	[43]
			DE/2-opt/1	2010	[44]
			DE/2-opt/2	2010	[44]
			DE/best/3	2011	[36]
		MDE-pBX	2012	[45]	
		IMDE	2013	[46]	
		MMS	2015	[47]	
		MPADE	2016	[48]	
		Fully directed	Trigonometric	2003	[30]
			ADE	2011	[49]
			Triangular	2015	[50]
			IDM	2015	[51]
AGDE	2017		[52]		
Probabilistic	GBDE	2013	[53]		
Novel Concept	ProDE	2011	[54]		
	Rank-DE	2013	[55]		
	NDI-DE	2013	[65]		
	Adaptive Greedy	2014	[57]		
	UDE	2016	[58]		
	FPS	2016	[59]		

b: DE/RAND/2

Price, Storn and Lampinen [33] proposed this scheme in order to obtain the mutant vector by choosing 5 mutually exclusive

vectors from the population to form 2 differences following the equation:

$$v_{i,g} = x_{r1,g} + F * (x_{r2,g} - x_{r3,g}) + F * (x_{r4,g} - x_{r5,g})$$

$$r1, r2, r3, r4, r5 \in [1, NP],$$

$$r1, \neq r2 \neq r3 \neq r4 \neq r5 \neq i \quad (6)$$

c: DE/CURRENT-TO-RAND/1

Mezura-Montes, Reyes, and Coello [34] presented a variant to generate the mutant vector using 3 mutually exclusive vectors that are different from the base vector. The 3 vectors with the base vector form 2 difference vectors that are added to the current base vector using the equation:

$$v_{i,g} = x_{i,g} + F * (x_{r1,g} - x_{i,g}) + F * (x_{r2,g} - x_{r3,g})$$

$$r1, r2, r3 \in [1, NP], r1, \neq r2 \neq r3 \neq i \quad (7)$$

d: NSDE (NEIGHBORHOOD SEARCH DIFFERENTIAL EVOLUTION)

Yang et al. [35] proposed a neighborhood search differential evolution based on the generalization of Neighborhood strategy (NS). Cauchy and Gaussian random numbers are used for long and small jumps, respectively. The mutant vector is generated using the equation:

$$v_{i,g} = x_{r1,g} + \begin{cases} d_{i,g} * N(0.5, 0.5) & \text{rand}_i(0, 1) < 0.5 \\ d_{i,g} * \delta & \text{otherwise} \end{cases}$$

$$d_{i,g} = x_{r2,g} - x_{r3,g} \quad (8)$$

where N(0.5,0.5) is a Gaussian random number with mean 0.5 and standard deviation 0.5,  $\delta$  is a Cauchy random number variable with scale parameter t = 1.

e: DE/RAND/3

Elsayed et al. [36] presented a framework that uses a pool of 4 mutations. The population is divided into for equal subpopulations, each subpopulation has its own individuals, and the value of scale parameter is self-adaptive. The mutant vector is obtained as follows:

$$v_i = x_{r1} + F * (x_{r2} - x_{r3} + x_{r4} - x_{r5} + x_{r6} - x_{r7}) \quad (9)$$

To calculate F, a Gaussian number N(0.5,0.15) is generated for each individual in the population.

$$F = F_{r1,G} + N(0, 0.5) * (F_{r2,G} - F_{r3,G}) + N(0, 0.5) * (F_{r4,G} - F_{r5,G}) + N(0, 0.5) * (F_{r6,G} - F_{r7,G}) \quad (10)$$

It must be mentioned that DE/rand/3 is proposed by Ting and Huang [37] without self-adaptive scaling factor.

f: DE/RAND-TO-CURRENT/2

This scheme is presented by Elsayed et al. [36]. The mutant vector is obtained as follows:

$$v_i = x_{r1} + F * (x_{r2} - x_i + x_{r3} - x_{r4}) \quad (11)$$



To calculate F, a Gaussian number  $N(0.5, 0.15)$  is generated for each individual in the population.

$$F = F_{r1,G} + N(0, 0.5) * (F_{r2,G} - F_{r3,G}) + N(0, 0.5) * (F_{r4,G} - F_{r5,G}) \quad (12)$$

#### g: DEGD (DIFFERENTIAL EVOLUTION WITH GENERALIZED DIFFERENTIALS)

Ali [38] proposed a new mutation strategy called DEGD, in which a mutant vector is generated only for worse solutions instead of generating trial vectors for all the population. DEGD generates more than one trial point per target vector (q points) until a successful trial point is found or the q number of trial points are reached. The first q-1 points are created using various values for  $F_1$  and  $F_2$  using the equation:

$$v = x_{r1} + F_1 * x_{r2} - F_2 * x_{r3} \quad (13)$$

while the qth point is found by vector projection as follows:

$$v = \left( \frac{x_{r1}^T * x_{r2}}{x_{r2}^T * x_{r2}} \right) * x_{r2} \quad (14)$$

where  $F_1 = F_2 = 0.5$ ,  $q = 3$

#### h: GPBX- $\alpha$

Dorrnsoro and Bouvry [39] proposed a new operator for generating the mutant vector called Gaussian PBX- $\alpha$  (GPBX- $\alpha$ ). The mutant vector is generated as follows:

$$\begin{aligned} v_{i,j} &= x_{r0,j} + \text{Gaussian}(0.0, 1.0) * F \\ &\quad * (UP_{i,j} - LOW_{i,j}) \\ UP_{i,j} &= \min(\max_j, x_{r0,j} - I_j * \alpha) \\ LOW_{i,j} &= \max(\min_j, x_{r0,j} + I_j * \alpha) \\ I_j &= x_{r1,j} - x_{r2,j} \end{aligned} \quad (15)$$

where Gaussian (0.0,1.0) is a random number from a Gaussian distribution centered at 0.0 with deviation 1.0,  $\max_j$  and  $\min_j$  are the upper and lower allowed bounds for variable j.

$$\alpha = \begin{cases} 0.2 + 0.6 * \text{rand}_1 & \text{if } \text{rand}_2 \leq 0.1 \\ \alpha & \text{otherwise} \end{cases} \quad F \in [0.1, 1.0] \quad (16)$$

#### i: IMMSADE

Wang *et al.* [40] proposed an improved version of the DE/rand/1 mode called IMMSADE, in which a new control parameter  $\omega_i$  that is associated with the base vector is introduced. And each individual in the population has its own  $\omega_i$  and  $F_i$ . The mutant vector is generated according to equation:

$$v_i^{t+1} = \omega_i * x_{r1}^t + F_i * (x_{r2}^t - x_{r3}^t), \text{ where } \omega_i \in [0.7, 1.0], F_i \in [0.1, 0.8] \quad (17)$$

## 2) PARTIALLY DIRECTED MUTATIONS

#### a: DE/CURRENT-TO-BEST/1

The scheme is presented by Price and Storn [1]. This version works typically like the DE/rand/1 except that there are

2 difference vectors in this version and also it incorporates the best vector found so far in order to enhance the greediness of the scheme. the mutant vector is generated according to the equation:

$$v_{i,g} = x_{i,g} + F * (x_{best,g} - x_{i,g}) + F * (x_{r1,g} - x_{r2,g}) \quad r1, r2 \in [1, NP], r1, \neq r2 \neq i \quad (18)$$

#### b: DE/BEST/1

Proposed by Price and Storn [33], in which two random vectors are used to generate the difference vectors. The base vector is chosen as the best individual.

$$v_{i,g} = x_{best,g} + F * (x_{r1,g} - x_{r2,g}) \quad r1, r2 \in [1, NP], r1, \neq r2 \neq i \quad (19)$$

#### c: DE/BEST/2

Price, Storn, and Lampinen [33] presented this scheme, like the DE/best/1 but with one additional difference vector.

$$v_{i,g} = x_{best,g} + F * (x_{r1,g} - x_{r2,g}) + F * (x_{r3,g} - x_{r4,g}) \quad r1, r2, r3, r4 \in [1, NP], r1, \neq r2 \neq r3 \neq r4 \neq i \quad (20)$$

#### d: DE/RAND-TO-BEST AND CURRENT/2

Presented by Elsayed *et al.* [36], the mutant vector is generated as follows:

$$v_i = x_{r1} + F * (x_{best} - x_{r2} + x_{r3} - x_i) \quad (21)$$

To calculate F, a Gaussian number  $N(0.5, 0.15)$  is generated for each individual in the population.

$$F = F_{r1,G} + N(0, 0.5) * (F_{r2,G} - F_{r3,G}) + N(0, 0.5) * (F_{r4,G} - F_{r5,G}) \quad (22)$$

It must be mentioned that DE/rand-to-best-and-current/2is proposed by Mezura-Montes *et al.* [41], but without self-adaptive scaling factor.

#### e: DE/RAND-TO-BEST/1

Referring to the DE literature. Qin *et al.* [42] developed this strategy that combines the strategies relying on the best solutions found so far and two-difference-vectors-based strategies in order to gain the benefits of those two strategies. The mutant vector is generated according to the equation:

$$v_{i,g} = x_{r1,g} + F * (x_{best,g} - x_{r1,g}) + F * (x_{r2,g} - x_{r3,g}) \quad r1, r2, r3 \in [1, NP], r1, \neq r2 \neq r3 \neq i \quad (23)$$

#### f: JADE

Zhang and Sanderson [43] proposed a new DE algorithm, JADE, by implementing a new mutation strategy DE/current-to-pbest/1 with the optional archive in order to improve the optimization process.

*g: DE/CURRENT-TO-PBEST/1 (WITHOUT ARCHIVE)*

The mutant vector is generated using the equation:

$$v_{i,g} = x_{i,g} + F_i * (x_{best,g}^p - x_{i,g}) + F_i * (x_{r1,g} - x_{r2,g}) \quad (24)$$

where  $x_{best,g}^p$  is randomly chosen as one of the top 100p% individuals in the population,  $p \in [0, 1]$ ,  $p = 5\%$ .  $F_i$ : mutation factor associated with each  $x_i$ , generated each generation using Cauchy distribution.  $F_i = randc(0.5, 0.1)$

*h: DE/CURRENT-TO-PBEST/1 (WITH ARCHIVE)*

A set of archived inferior solutions ‘‘A’’ that contains the Recently explored inferior solutions, when compared to the current population ‘‘P’’, is used in order to provide additional information about the promising progress dimension. The mutant vector is generated as follows:

$$v_{i,g} = x_{i,g} + F_i * (x_{best,g}^p - x_{i,g}) + F_i * (x_{r1,g} - x'_{r2,g})$$

$$F_i = randc(0.5, 0.1), \text{cauchydistribution} \quad (25)$$

where  $x_{i,g}, x_{r1,g}, x_{best,g}^p$  are selected from P while  $x'_{r2,g}$  is randomly chosen from the union  $P \cup A$  of the current population and archive. The archive starts empty, and then filled with the parent who failed in selection in each generation. If the archive exceeds the size, some elements are deleted from it in order to keep the max size of the archive.

$$CR = randn(0.5, 0.1), \text{normaldistribution}$$

*i: DEGL: DE USING A NEIGHBORHOOD-BASED MUTATION OPERATOR*

Das et al. [20] proposed a new variant that is based on utilizing the concept of the neighborhood of each individual in the population. The idea based on that there is a DE population  $P_G = [x_{1,G}, x_{2,G}, \dots, x_{NP,G}]$  at generation G. The vector indices are sorted randomly for every  $x_{i,G}$ . They define a neighborhood of radius  $k$ ,  $k > 0$ ,  $k \in [0, (NP - 1)/2]$  consisting of vectors  $x_{i-k,G}, \dots, x_{i,G}, \dots, x_{i+k,G}$ . The vectors are organized on a ring topology with respect to their indices, such that vectors  $x_{NP,G}$  and  $x_{2,G}$  are the two immediate neighbors of the vector  $x_{1,G}$ . Then, for every member in the population, calculate

Local donor	$L_{i,G} = x_{i,G} + \alpha * (x_{n-best} - x_{i,G}) + \beta * (x_{p,G} - x_{q,G})$ <p><math>x_{n-best}</math> : the best vector in the neighborhood.  <math>p, q \in [i - k, i + k] p \neq q \neq i</math></p>
Global donor	$g_{i,G} = x_{i,G} + \alpha * (x_{g-best} - x_{i,G}) + \beta * (x_{r1,G} - x_{r2,G})$ <p><math>x_{g-best}</math> : the best vector in the entire population at generation G.  <math>r1, r2 \in [1, NP] r1 \neq r2 \neq i</math></p>
$\alpha, \beta$ : scaling factors	

The mutant vector is generated using the equation

$$v_{i,G} = w_{i,G} * g_{i,G} + (1 - w_{i,G}) * L_{i,G}, w_{i,G} \approx rand[0, 1] \quad (26)$$

With  $k = 10\% * NP$  and  $\alpha = \beta = 0.8$ .

*j: A 2-OPT BASED DIFFERENTIAL EVOLUTION FOR GLOBAL OPTIMIZATION*

Chiang et al. [44] proposed DE/2-opt/1 and DE/2-opt/2 in order to overcome the problem of long computational time of DE.

*k: DE/2-OPT/1*

$$v_i = \begin{cases} x_{r1} + F * (x_{r2} - x_{r3}) & \text{if } f(x_{r1}) < f(x_{r2}) \\ x_{r2} + F * (x_{r1} - x_{r3}) & \text{otherwise} \end{cases} \quad (27)$$

*l: DE/2-OPT/2*

$$v_i = \begin{cases} x_{r1} + F * (x_{r2} - x_{r3}) + F * (x_{r4} - x_{r5}) & \text{if } f(x_{r1}) < f(x_{r2}) \\ x_{r2} + F * (x_{r1} - x_{r3}) + F * (x_{r4} - x_{r5}) & \text{otherwise} \end{cases} \quad (28)$$

$r_1, r_2, r_3, r_4, r_5$  are mutually exclusive and picked randomly from the population and are different from  $i$ .

Where  $F = 0.5$

*m: DE/BEST/3*

Presented by Elsayed et al. [36]

$$v_i = x_{best} + F * (x_{r2} - x_{r3} + x_{r4} - x_{r5} + x_{r6} - x_{r7}) \quad (29)$$

To calculate F, a Gaussian number N (0.5,0.15) is generated for each individual in the population.

$$F = F_{r1,G} + N(0, 0.5) * (F_{r2,G} - F_{r3,G}) + N(0, 0.5) * (F_{r4,G} - F_{r5,G}) + N(0, 0.5) * (F_{r6,G} - F_{r7,G}) \quad (30)$$

*n: MDE-pBX*

Islam et al. [45] proposed MDE-pBX algorithm with a new mutation operator, DE/current-to-gr\_best/1, that uses an individual from the best q% individuals in the current population to generate the mutant vector according to the equation:

$$v_{i,g} = x_{i,g} + F_i * (x_{gr-best,g} - x_{i,g} + x_{r1^i,g} - x_{r2^i,g}) \quad (31)$$

where  $x_{gr-best,g}$  is the best of the q% vectors randomly chosen from the current population.  $x_{r1^i,g}$  and  $x_{r2^i,g}$  are two distinct vectors picked up randomly from the current population and none of them is equal to  $x_{gr-best,g}$  or the target vector. The authors used  $q = 15\%$  and  $F_i = Cauchy(0.5, 0.1)$ .

*o: IMDE*

Zhou et al. [46] proposed a novel mutation called intersect mutation differential evolution (IMDE), in which all individuals are ranked from worse to better according to their fitness value. Then the population is divided into 2 main parts (better part and worse part).

*p*: FOR THE BETTER PART

The mutant vector is generated using one individual from the worst part and two individuals were chosen from the best part using the equation:

$$v_{i,g+1} = x_{wr1,g} + F * (x_{br1,g} - x_{br2,g})$$

$$br1 \neq br2 \neq wr1 \neq i \quad (32)$$

where  $x_{wr1,g}$  is an individual from the worse part,  $x_{br1,g}, x_{br2,g}$  are two individuals from the better part.

*q*: FOR THE WORSE PART

The mutant vector is generated using one individual from the best part and two individuals from the worse part using the equation:

$$v_{i,g+1} = x_{br1,g} + F * (x_{wr1,g} - x_{wr2,g})$$

$$br1 \neq wr1 \neq wr2 \neq i \quad (33)$$

where  $F = 0.5$

*r*: MMS

Ali et al. [47] presented a novel mutation strategy called (MMS) that uses the information from either the best or a randomly selected individual in order to increase the quality of solutions. In this method, the choice of base vector is different from DE. The mutant vector is generated according to the following equations:

If  $(x_{best}(t) - x_{best}(t - 1)) < \epsilon$ , where  $\epsilon = 1 * 10^{e-6}$

$$v = (a_1 x_{bestj}(t) + a_2 x_{r1,j}(t) + a_3 x_{r2,j}(t))$$

$$+ F * (x_{r1,j}(t) - x_{r2,j}(t)) \quad (34)$$

Else

$$v = (a_1 x_{r1,j}(t) + a_2 x_{r2,j}(t) + a_3 x_{r3,j}(t))$$

$$+ F * (x_{r1,j}(t) - x_{r2,j}(t)) \quad (35)$$

where  $a_1, a_2, a_3$  are chosen randomly from the interval  $[0,1]$ ,  $\sum_a = 1, F \in [0.5, 0.9]$

*s*: MPADE

Cui et al. [48] presented a novel adaptive multiple sub-population based DE algorithm named MPADE, that is inspired by the concept of work specialization. the algorithm divides the entire population into several sub-groups that are responsible for different tasks due to their capabilities.

**First**, all individuals are sorted based on fitness.

**Second**, the population is divided into 3 sub-populations

- o Inferior sub-pop of size:  $w_1 * NP$
- o Medium sub-pop of size:  $w_2 * NP$
- o Superior sub-pop of size:  $w_3 * NP$

Where  $w_i \in [0, 1]$  and  $\sum_{i=1}^3 w_i = 1$

**Third**, each individual selects ns closest individuals and rs farthest individuals based on the Euclidean distance

$$ns = \frac{NP}{10} + ceil\left(\frac{2NP}{5} * \left(1 - \frac{g-1}{G_{max}}\right)\right) \quad (36)$$

$$rs = \frac{NP}{10} + ceil\left(\frac{2NP}{5} * \left(\frac{g-}{G_{max}}\right)\right) \quad (37)$$

For the **inferior subpopulation**, the mutant vector is generated as follows:

$$v_i = x_i + F_i * (x_{best} - x_i) + F_i * (x_{r1} - x_{r2})$$

$$+ F_i * (x_{r3} - x_{r4}) \quad (38)$$

where,  $x_{best}$  is the best individual among the relatives of the target vector.  $x_{r1}, x_{r2}, x_{r3}, x_{r4}$  are randomly chosen from the current population.

For the **superior subpopulation**, the mutant vector is generated as follows:

$$v_i = x_i + F_i * (x_{nbest} - x_i) + F_i * (x_{r1} - x_{r2})$$

$$+ F_i * (x_{r3} - x_{r4}) \quad (39)$$

where,  $x_{nbest}$  is the best individual among the relatives of the target vector.

For the **medium sub population**, the mutant vector is generated as follows:

$$v_i = x_i + F_i * (x_{pbest} - x_i) + F_i * (x_{r1} - x_{r2})$$

$$+ F_i * (x_{r3} - x_{r4}) \quad (40)$$

where,  $x_{pbest}$  is the best individual of the ps vectors that are randomly selected from the current population.

$$F_i = cauchy(0.5, 0.1)$$

3) FULLY DIRECTED MUTATIONS

*a*: TRIGONOMETRIC MUTATION (TDE)

Developed by Fan and Lampinen [30] to increase the convergence velocity of the DE. In this methodology, a new local search operation is introduced in order to speed up the DE when optimizing expensive objective functions. The mutant vector is generated according to the equation:

$$v_{i,g+1} = \left\{ \begin{array}{l} \frac{[x_{r1,g} + x_{r2,g} + x_{r3,g}]}{3} \\ + (p_2 - p_1) * (x_{r1,g} - x_{r2,g}) \\ + (p_3 - p_2) * (x_{r2,g} - x_{r3,g}) \\ + (p_1 - p_3) * (x_{r3,g} - x_{r1,g}) \end{array} \right\} \quad (41)$$

where:

$$p_1 = \frac{|f(x_{r1,g})|}{p'}, p_2 = \frac{|f(x_{r2,g})|}{p'}, p_3 = \frac{|f(x_{r3,g})|}{p'}$$

$$p' = |f(x_{r1,g})| + |f(x_{r2,g})| + |f(x_{r3,g})|$$

$$r1, r2, r3 \in [1, NP], r1, \neq r2 \neq r3 \neq i$$

*b*: ADE

Mohamed et al. [49] presented an alternative differential evolution algorithm (ADE). The new directed mutation scheme is proposed based on the weighted difference between the best and worst individuals at a particular generation. The mutant vector is generated as follows:

$$v_i^{g+1} = x_r^g + F * (x_b^g - x_w^g) \quad (42)$$



where  $x_r$  is chosen randomly from the population at generation  $g$ ,  $x_b$  is the best individual at generation  $g$  and  $x_w$  is the worst individual at generation  $g$ .  $F$  is a uniform random number between  $[0, 1]$ .

*c: TRIANGULAR*

Mohamed [50] proposed a novel mutation called triangular mutation. In which, three vectors are randomly selected. The three vectors are sorted ascendingly due to their objective function values to get  $x_{best,j}$ ,  $x_{better,j}$ ,  $x_{worst,j}$ . The mutant vector is generated according to the equation:

$$v_{i,j}^{g+1} = \bar{x}_{c,j}^g + F_i * (x_{best,j}^g - x_{better,j}^g) + F_i * (x_{better,j}^g - x_{worst,j}^g) + F_i * (x_{best,j}^g - x_{worst,j}^g) \quad (43)$$

where:

$F_i$  is a uniform random number between  $[0.2, 0.8]$

$$\bar{x}_{c,j}^g = w_1 * x_{best} + w_2 * x_{better} + w_3 * x_{worst}$$

$$w_i \geq 0, \sum_{i=1}^3 w_i = 1$$

$$w_i = \frac{P_i}{\sum_{i=1}^3 P_i}, i = 1, 2, 3$$

$$P_1 = 1, P_2 = rand(0.75, 1) \text{ and } P_3 = rand(0.5, P_2)$$

*d: IDM: INDIVIDUAL DEPENDENT MUTATION*

Tang et al. [51] proposed a novel variant of DE with an individual dependent mechanism. In the mechanism the population is divided into two non-overlapping sets: superior and inferior, the superior proportion is in the equation  $p_s$ , and the mutant vector is generated according to the equation:

$$v_{i,g} = \left\{ \begin{array}{l} x_i + F * (x_{r1} - x_i) + F * (x_{r2} - d_{r3}) \\ i \in superior \\ x_i + F * (x_{better} - x_i) + F * (x_{r2} - d_{r3}) \\ i \in inferior \end{array} \right\} \quad (44)$$

where  $F = 0.5$ ,  $x_{better}$  is randomly selected from a set superior.

$$d_{r3,g}^j = \left\{ \begin{array}{ll} L^j + rand(0, 1) * (U^j - L^j) & rand(0, 1) < p_d \\ x_{r3}^j & otherwise \end{array} \right\}$$

$$p_d = 0.1 * p_s$$

$$p_s = 0.1 + 0.9 * 10^{5 * (\frac{g}{g_{max}} - 1)} \quad (45)$$

*e: AGDE*

Mohamed and Mohamed [52] presented a novel mutation scheme by utilizing the information on good and bad individuals in the population, the algorithm named AGDE. In each generation, the population is divided into three clusters (best, better, and worst) of sizes  $100p\%$ ,  $NP-2*(100p\%)$  and  $100p\%$  respectively. Three vectors are selected randomly, one from each partition to generate the mutant vector based on the

following equation:

$$v_i^{g+1} = x_r^g + F * (x_{p-best}^g - x_{p-worst}^g) \quad (46)$$

where  $x_r^g$  is chosen randomly from the middle  $NP-2*(100p\%)$ ,  $x_{p-best}^g, x_{p-worst}^g$  are chosen randomly from the top and bottom  $100p\%$ , where  $p = 10\%$ ,  $F$  is a uniform random number between  $[0.1, 1]$ .

4) PROBABILISTIC MUTATIONS

*a: GBDE*

Wang et al. [53] proposed a new mutation operator called Gaussian Bare-Bones DE (GBDE). The mutant vector is generated by a Gaussian distribution based on the current and the best individual at the current generation as follows:

$$v_{i,g} = N(\mu, \sigma) \quad (47)$$

where  $N(\mu, \sigma)$  is a Gaussian random function with mean  $\mu$  and standard deviation  $\sigma$

$$\mu = \frac{x_{best,g} - x_{i,g}}{2}$$

$$\sigma = |x_{best,g} - x_{i,g}|$$

The algorithm is Explorative at the start, but as the generation increases the difference between the best and any individual will decrease, and the average will go toward the best.

**B. NOVEL CONCEPT**

*b: PRODE*

Epitropakis et al. [54] proposed a novel framework called proximity-based DE (ProDE), in which neighbors of a parent vector, rather than the random ones will be used to generate the donor vector. The framework consists of 3 steps to generate the mutant vector as follows:

- Compute the pair-wise distance between all members of the population.

$$R = [r_{ij}]_{NP * NP} \quad (48)$$

where  $r_{ij}$  is the distance between the  $i$ th and the  $j$ th members of the population.

- Probability matrix is calculated:

$$R_p(i,j) = 1 - \frac{r_{ij}}{\sum_{k=1}^{NP} r_{ik}} \quad (49)$$

where the minimum distant neighbor of a vector will have the highest probability to be selected as  $r_i$  index.

- Select three vectors with indices  $r_1, r_2$  and  $r_3$  based on the  $R_p(i, :)$

$$v_i = x_{r1} + F * (x_{r2} - x_{r3}) \quad (50)$$

Then update  $R$  and  $R_p$  for the new offspring.

c: RANK-DE

In nature, good species always contain valuable information. Gong and Cai [55] proposed a new idea, Rank-DE, inspired by the phenomenon. Where parents are proportionally selected according to their ranks in the current population. The algorithm based on 3 steps:

- The population is Sorted based on the fitness of each individual.
- A selection probability is calculated for each individual

$$P_i = \frac{R_i}{NP}, i = 1, 2, \dots, NP$$

where  $R_i = NP - i$

- Two individuals  $x_{r1}, x_{r2}$  are selected based on their selection probability, and the third one is selected randomly from the population  $x_{r3}$  in order to generate the mutant vector.

$$v_{i,g} = x_{r1} + F * (x_{r2} - x_{r3}) \quad (51)$$

d: NDI-DE

Cai and Wang proposed [56] a new idea based on the neighborhood and direction called neighborhood and direction information-based DE (NDI-DE). The mutant vector is generated as follows:

$$v_{i,g} = x_{base,g} + F * x_{diff,g} + DT_{i,g}, \dots, F = 0.5 \quad (52)$$

- For calculating  $x_{base}$  and  $x_{diff}$ :

Based on the neighborhood information, a probability to each vector of the population is generated based on the Euclidean distance from the target individual I for any vector j

$$P_j = 1 - \frac{\|x_i, x_j\|}{\sum_{j=1}^{NP} \|x_i, x_j\|} \quad (53)$$

A roulette wheel selection is used for selecting  $x_{r1}, x_{r2}, x_{r3}$  based on the probability values in the previous equation.

After the selection of  $x_{r1}, x_{r2}, x_{r3}$ , the winner of the tournament is  $x_{base}$  and the other 2 vectors will form the  $x_{diff}$

- For calculating DT (the direction):

One of the next 3 cases is chosen (each is tested separately to check its effect):

DA: directional attracted, derived from the best near neighbor individual.

$$DA_i = I_{DA} * (x_{ibest} - x_i) \quad (54)$$

where  $x_{ibest}$  is the best individual nearest neighbor

DR: directional repulsion, derived from the worst near neighbor individual

$$DR_i = -I_{DR} * (x_{iworst} - x_i) \quad (55)$$

where  $x_{iworst}$  is the worst individual nearest neighbor.

DC: directional convergence, derived from the combination of the best and worst near neighbor individuals

$$DC_i = I_{DC1} * (x_{ibest} - x_i) - I_{DC2} * (x_{iworst} - x_i) \quad (56)$$

where  $I_{DA}, I_{DR}, I_{DC}$  are scaling factors.

e: ADAPTIVE GREEDY

Yu et al. [57] introduced new mutation strategies that utilize the information of top t individuals in the current population. The parent vector is randomly selected from the top t solutions in the current population. The new strategies are:

- DE/atbest/1

$$v_{i,g} = x_{best,g}^t + F * (x_{r1,g} - x_{r2,g}) \quad (57)$$

- DE/atbest/2

$$v_{i,g} = x_{best,g}^t + F * (x_{r1,g} - x_{r2,g}) + F * (x_{r3,g} - x_{r4,g}) \quad (58)$$

- DE/current-to-atbest/1

$$v_{i,g} = x_{i,g} + F * (x_{best,g}^t - x_{i,g}) + F * (x_{r1,g} - x_{r2,g}) \quad (59)$$

Each individual in the population is associated with a value of t, that is selected randomly between [1, NP]. Where  $F = 0.5$ .

f: UDE

Sharifi-Noghabi et al. [58] presented a Union Differential Evolution (UDE) to intelligently select the mutation vectors considering the advantages of both design and fitness spaces criteria. The mutant vector is generated as follows:

$$v_i = x_{FS1} + F * (x_{FS2} - x_{r1}) + F * (x_{DS} - x_{r2}) \quad (60)$$

where  $x_{r1}, x_{r2}$  are selected randomly from the population.

$x_{FSi}$  is the parent vector chosen by fitness space criterion, to obtain the  $x_{FSi}$ :

- Sort the population in increasing order (from best to worst) due to fitness value.
- Calculate the selection probability for each individual

$$P_i = \frac{NP - i}{NP}, i = 1, 2, \dots, NP \quad (61)$$

- Select 2 members  $x_{FS1}, x_{FS2}$  using roulette wheel.

$x_{DS}$  is the vector selected by design space criterion, to obtain  $x_{DS}$ :

- Based on the Euclidean distance between all the individuals in the population, the distance matrix DM is:

$$DM = \begin{pmatrix} \|x_1 - x_1\| & \dots & \|x_1 - x_{NP}\| \\ \vdots & \ddots & \vdots \\ \|x_{NP} - x_1\| & \dots & \|x_{NP} - x_{NP}\| \end{pmatrix} \quad (62)$$

- Based on DM, calculate the probability matrix PM

$$PM(i, j) = 1 - \frac{DM(i, j)}{\sum_{\forall k} DM(i, k)} \quad (63)$$

- Roulette wheel selection without replacement is performed on every row of PM matrix (for each member of the population) in order to choose  $x_{DS}$ .

$g$ : FPS

Cai *et al.* [59] presented a new selection method called, fitness and position-based selection (FPS). The new method utilizes the population information in order to select individuals for mutation. The method has 3 steps and applied on 6 mutation strategies from the literature.

*Step 1:* Calculate the influence value of each individual

$$\begin{aligned} INF_i(x_{j,g}) &= \frac{FPR_i(x_{j,g})}{\sum_{k=1}^{NP} \text{and } k \neq i FPR_i(x_{k,g})} + \varepsilon, \varepsilon = 0.001 \\ FPR_i(x_{j,g}) &= \frac{M_f(x_{j,g})}{M_{d,j}(x_{j,g})} \\ M_f(x_{j,g}) &= |f(x_{j,g}) - f(x_{worst,g})| \\ M_{d,i}(x_{j,g}) &= \|x_{i,g}, x_{j,g}\| \end{aligned} \quad (64)$$

where,  $x_{worst,g}$  is the worst individual in the current population,  $\|a, b\|$  is the Euclidean distance between  $a$  and  $b$ .

*Step 2:* Calculate the selection probability of each individual

$$P_{i,j,g} = \frac{INF_i(x_{j,g})}{\sum_{k=1}^{NP} INF_i(x_{k,g})} \quad (65)$$

*Step 3:* Roulette wheel selection is made in order to select the parents based on the probability.

Then, the mutant vector  $v_{i,g}$  is calculated using 6 mutation schemes from the literature:

$$\begin{array}{ll} FPS-DE/rand/1 & FPS-DE/rand/2 \\ FPS-DE/best/1 & FPS-DE/best/2 \\ FPS-DE/current-to-best/1 & FPS-DE/rand-to-best/1 \end{array}$$

#### IV. NUMERICAL EXPERIMENTS AND COMPARISONS

In fact, there is an important question that needs to precise and concise answer. How much improvement could be achieved by the proposed mutations? Thus, in order to answer this question and as a guideline for researchers, practitioners and interested scientists, the best mutations must be identified for further improvement as well as the worst mutations must be also identified for further investigation with the possible repair. Consequently, in this section, to evaluate the performance of all mutations, the computational results of all mutations along with appropriate statistical analysis are discussed.

##### A. EXPERIMENTS SETUP

The performance of the proposed DE-based algorithms using these mutations was tested on 30 benchmark functions proposed in the CEC 2017 special session on real-parameter optimization. A detailed description of these test functions can be found in [29]. These 30 test functions can be divided into four classes:

- Unimodal functions  $f1 - f3$ ;
- Simple multimodal functions  $f4 - f10$ ;
- Hybrid functions  $f11 - f20$ ;
- Composition functions  $f21 - f30$ .

Note that  $f2$  has been excluded because it shows unstable behavior especially for higher dimensions.

##### B. PARAMETER SETTINGS

To evaluate the performance of these algorithms using the proposed mutations, experiments were conducted on the test suite. We adopt the solution error measure ( $f(x) - f(x^*)$ ), where  $x$  is the best solution obtained by algorithms in one run and  $x^*$  is the well-known global optimum of each benchmark function. Error values and standard deviations smaller than  $10^{-8}$  are taken as zero. The dimensions ( $D$ ) of function are 10, 30, 50 and 100, respectively. The maximum number of function evaluations (FEs), the terminal criteria, is set to  $10000 \times D$ , all experiments for each function and each algorithm run 51 times independently. Besides, for fair comparison and to test the individual effect of the mutation on the optimization process, adaptation and/or self-adaptation schemes for control parameters are not allowed and disabled. Thus, the population size NP is set to 100. The binomial crossover operator is utilized in all mutations strategies due to its popularity in many DE literatures [33], [42], as shown in (3). The crossover factor ( $CR$ ) is set to 0.9. However, regarding scaling factor ( $F$ ), it is considered as a part of the mutation itself. Thus, as aforementioned in section III, the constant value of 0.5 or random values using a specific probability distribution as mentioned in the original algorithm are allowed. The presentation of the experimental results is divided into two subsections. First, an empirical taxonomy based on an overall performance and comparison between all mutations over all dimensions is provided. Second, to compare and analyze the solution quality from a statistical angle of different algorithms and to check the behavior of the stochastic algorithms (García *et al.*) [60], the results are compared using two non-parametric statistical hypothesis tests: (i) the Friedman test (to obtain the final rankings of different algorithms for all functions). (ii) multi-problem Wilcoxon signed-rank test (to check the differences between all algorithms for all functions); at a 0.05 significance level, where  $R^+$  denotes the sum of ranks for the test problems in which the first algorithm performs better than the second algorithm (in the first column), and  $R^-$  represents the sum of ranks for the test problems in which the first algorithm performs worse than the second algorithm (in the first column). Larger ranks indicate larger performance discrepancy. As a null hypothesis, it is assumed that there is no significance difference between the mean results of the two samples. Whereas the alternative hypothesis is that there is significance in the mean results of the two samples, the number of test problems  $N = 29$  for  $D = 10, 30, 50$  and 100 dimensions and 5% significance level. Use the p-value and compare it with the significance level. Reject the null hypothesis if the p-value is less than or equal the significance level (5%). All the p values in this paper were computed using SPSS (the version is 20.00). Third, the convergence behavior of top ten algorithms is analyzed. Then, the performance of all algorithm

that provided minimum mean function error on the CEC 2017 functions with all dimensions is presented.

### C. EXPERIMENTAL RESULTS AND COMPARISON

The statistical results of all algorithms on the benchmarks with 10, 30, 50 and 100 dimensions are summarized in the supplemental file (Tables S1-S30). It includes the obtained mean and the standard deviations of error from the optimum solution of all algorithms over 51 runs for all 29 benchmark functions.

#### 1) EMPIRICAL TAXONOMY AND COMPARISON

In order to analyze and compare all results provided by all algorithms, ranking of the algorithms on the CEC 2017 functions with 10, 30, 50 and 100 dimensions are given in Figure 3. Firstly, it can be observed from Figure 3 that most of the algorithms perform differently on different dimensions. Generally, according to the growth of the search-space dimensionality from 10D to 100D, the performance of all algorithms can be classified into four main categories as depicted in Figure 4. The first category includes algorithms with excellent performance that show outstanding continuous improvement as the dimension of the search-space increases i.e., it has a better rank in 100D than its rank in 10D with continuous improvement. The second category includes algorithms show unstable performance and slight diminishes as dimensions of the functions increases i.e., it has good rank in 10D with good performance, but its performance slightly fluctuates with slight deterioration with 30D, 50D, and 100D. The third category includes algorithms show descent performance with complete and/or significant deterioration with the growth of the search-space dimensionality i.e., it has a better rank in 10D than its rank in 100D with continuous diminishes. The fourth category includes algorithms show stable moderate or poor performance with insignificant improvement in all dimensions and/or they get the lower ranking in most cases. Actually, it can be obviously shown from Fig.3 that AGDE belongs to the first category as it gets the sixth, first, second and first ranking in 10D, 30D, 50D and 100D, respectively. Therefore, AGDE shows perfect performance with continuous improvement as the dimension of the functions increases. it is still more stable, efficient and robust against the curse of dimensionality. Besides, JADE, GBPX, and MPADe also get 11th, 16th and 14th ranking in 10D, 6th, 13th, 10th ranking in 30D, 6th, 8th, 11th ranking in 50D and 2nd, 4th, 6th ranking in 100D followed by ADE, DE/best/3 and DEGL although they get lower ranking, they follow the same pattern. Furthermore, the performance of AGDE, JADE, GBPX and MPADe algorithms does not affect by switching off self-adaptive mechanism of crossover and scaling factor that has been used along with the proposed novel mutation. Thus, it is noteworthy to mention that these mutations represent role model mutations as they still have outstanding performance during the optimization process. Due to its performance with slight fluctuations and deteriorations, it can be obviously seen from Figure 3 that DE/rand-

to/current/2, DE/rand-to-best and current/2, DE/rand/3, Triangular, JADE without an archive, DE/current-to-rand/1, DE/best/2 and MMS belong to the second category. Besides, on the contrary of the first category, the third category includes all algorithm with complete deterioration as dimensions increases such NSDE, DE/2-opt/1, DEGD, IMDE, DE/rand/1 and MDE-pBX algorithms. For instance, it can be deduced from Figure 3 that NSDE, DE/2-opt/1, and DEGD get 1st, 3rd, and 2nd ranking in 10D, 8th, 9th and 7th ranking in 30D, 9th, 10th and 7th ranking in 50D and 10th, 11th and 9th ranking in 100D. Finally, regarding the fourth category, GPDE, Trigonometric, and DE/best/1 were the weaker in performance in addition to the remaining algorithms DE/rand/2, DE/current-to-best/1, DE/2-opt/2, DE/rand-to-best/1, IMMSADE and IDM get lower ranking with the most of dimensions.

#### 2) STATISTICAL ANALYSIS

On the other hand, Table 2 lists the Average ranks for all algorithms across all problems and all dimensions according to Friedman test. The best ranks are shown in bold and the second ranks are underlined. Besides, the rank of all algorithms on the CEC 2017 functions is shown in Figure 5. The p-value computed through Friedman test is 0.00E+00. Thus, it can be concluded that there is a significant difference between the performances of the algorithms. It can be clearly seen from Table 2 that, regarding mean ranking, AGDE gets the first ranking followed by DE/rand-to/current/2, DE/rand-to-best, and current/2 and JADE gets fourth ranking. Taking into consideration that regarding ranking DE/rand-to/current/2, DE/rand-to-best and current/2 perform better than JADE in 30D and 50D while JADE outperforms DE/rand-to/current/2, DE/rand-to-best and current/2 in 100D.

Due to its outstanding performance and being first ranking, the multi-problem Wilcoxon signed-rank test between AGDE and other algorithms in 10D, 30D, 50D and 100D are summarized in Tables III,IV,V and VI, respectively. From Table 3, we can see that AGDE obtains higher R+ values than R- in most of the cases, while slightly lower R+ value than R-value in comparison with JADE, JADE with no archive, MDE-pBX, DE/rand-to-best and current/2 and MMS algorithms. However, in the cases of AGDE versus DE/2-opt/1, NSDE, IMDE, DEGD and DE/rand/1, they get higher R- than R+ values. The reason is that AGDE gains the performance far away of what these five algorithms do on some functions, resulting in higher ranking values. According to the Wilcoxon's test at  $\alpha = 0.05$ , the significant difference can be observed in 11 cases, which means that AGDE is significantly better than 9 algorithms out of 29 algorithms on 29 test functions while it is significantly outperformed by NSDE and DEGD algorithms. However, there is no significant difference in the remaining 18 cases. Regarding 30D, 50D and 100D problems, the results of multi-problem Wilcoxon's test in Tables IV, V and VI shows that AGDE obtains higher R+ values than R- in all cases with exception of DE/rand-to-current/2 algorithms in  $D = 50$ , and JADE and MPADe



algorithms in  $D = 100$ . According to the Wilcoxon’s test at  $\alpha = 0.05$ , the significance difference can be observed in 23, 21 and 16 cases in  $D = 30, 50$  and  $100$ , respectively, while there is the insignificant difference in all remaining cases. Thus, the performance of AGDE is always better than or equal to other compared algorithms in  $D = 30, 50$  and  $100D$ . Alternatively, to be more precise, it is obvious from Tables III-VI that AGDE is inferior to, equal to, superior to other algorithms in 227,109, 505 out of the total 841 cases in 10D, 133, 41, 667 out of the total 841 cases in 30D, 167, 5, 669 out of the total 841 cases in 50D, 177,4,660 out of the total 841 cases in 100D, respectively. In summary, AGDE is inferior to, equal to, superior to other algorithms in 704, 159, 2501 cases, respectively out of total 3364 cases. Note that a total number of cases is given such that (29 algorithms \* 29 problems \* 4 dimensions = 3364). Thus, it can be concluded that the performance of AGDE is almost better than the performance of compared algorithms in 74.4 % of all cases, respectively, and it is just outperformed by other compared algorithms in 20.9 % of all problems in all dimensions. Furthermore, it can be obviously deduced from Fig.6 that the superiority of the AGDE algorithm against the compared algorithms increases as the dimensions of the problems increases from 10 to 100 dimensions.

### 3) CONVERGENCE BEHAVIOR ANALYSIS

Furthermore, convergence behavior is another important factor that must be considered in comparison among all proposed algorithms. Therefore, for better presentation, the convergence behavior of top ten algorithms will be analyzed. Thus, the convergence characteristics in terms of the best fitness value of the median run of all algorithms for some functions with dimensions 10,30, 50 and 100 is illustrated in the supplemental file (Figure S1). Regarding Unimodal functions  $f1 - f3$ , From Figure S1, it can be observed that JADE covered to better solutions faster than all other algorithms. Moreover, it provides better solutions than other compared algorithms especially on  $D = 50$  and  $100$ . Concerning the remaining functions, Simple multimodal functions  $f4 - f10$ , Hybrid functions  $f11 - f20$ ; and Composition functions  $f21 - f30$  it can be obviously seen that, on the same problem, most of the algorithms converge differently on different dimensions. JADE provides better solutions than other compared algorithms in  $f12$  on  $D = 50$ ,  $f18$  on  $D = 50,100$ , and  $f19$  on  $D = 100$ . AGDE shows superior performance in solving problems  $f11$  and  $f14$  on  $D = 10$ ,  $f14$  on all dimensions,  $f17$  on  $D = 50$  and  $100$ ,  $f20$  on  $D = 50$ ,  $f21$  on  $D = 50$  and  $100$ ,  $f26$  on  $D = 30$ . Generally, it is clear that the convergence speed of the majority of these algorithms is fast in the early stage of the optimization process for all functions with different shapes, complexity, and dimensions. Furthermore, the convergence speed is dramatically decreased, and its improvement is found to be significant in the middle and later stages of the optimization process. Additionally, the convergent figure suggests that many algorithms can reach the global solution or better solution in most problems in a fewer

**TABLE 2. Average ranks for all algorithms across all problems and all dimensions according to friedman test.**

Algorithm	10D	30D	50D	100D	Mean Ranking	Rank
AGDE	10.83	<b>6.29</b>	<u>6.84</u>	7.34	<b>7.83</b>	<b>1</b>
DE/rand-to-current/2	11.16	8.45	<b>5.97</b>	7.69	<u>8.31</u>	2
DE/rand-to-best and current/2	11.03	7.47	7.59	9.45	8.88	3
JADE	11.33	9.40	9.69	<u>7.43</u>	9.46	4
Triangular	12.33	<u>6.71</u>	9.05	10.16	9.56	5
DEGD	<u>8.28</u>	9.52	10.03	10.59	9.60	6
DE/rand/3	12.84	8.86	7.66	10.12	9.87	7
NSDE	<b>7.09</b>	10.12	11.21	11.53	9.99	8
DE/2-opt/1	8.53	10.22	11.52	11.66	10.48	9
MPADE	14.33	11.47	11.62	9.91	11.83	10
GBPX- $\alpha$	14.98	12.55	11.02	8.88	11.86	11
JADE with no archive	11.22	11.62	13.02	12.16	12.00	12
IMDE	9.38	12.91	12.88	13.14	12.08	13
DE/rand/1	9.36	12.91	12.69	13.71	12.17	14
MDE-pBX	11.16	12.48	14.29	13.26	12.80	15
DEGL	15.40	12.78	12.48	12.29	13.24	16
MMS	14.79	17.10	16.31	14.93	15.78	17
DE/best/3	21.16	17.26	15.83	14.57	17.20	18
DE/best/2	17.97	17.03	17.74	17.72	17.62	19
DE/current-to-rand/1	15.22	18.95	19.62	19.05	18.21	20
IDM	15.69	18.91	19.14	19.12	18.22	21
DE/2-opt/2	15.91	20.81	20.55	21.55	19.71	22
DE/rand-to-best/1	20.26	20.17	20.02	21.19	20.41	23
IMMSADE	21.93	22.02	19.98	20.12	21.01	24
ADE	22.69	21.55	20.69	19.83	21.19	25
DE/current-to-best/1	20.21	21.29	21.40	22.45	21.34	26
DE/rand/2	17.86	22.83	23.28	23.62	21.90	27
DE/best/1	25.50	24.93	24.66	24.17	24.81	28
Trigonometric	28.84	28.72	28.84	28.50	28.73	29
GBDE	27.72	29.66	29.40	28.86	28.91	30

number of generations less than the maximum predetermined number of generations. In general, the majority of the top ten algorithms are scalable enough and can balance greatly the exploration and exploitation abilities until the maximum FEs is reached.

### 4) SUPERIOR PERFORMANCE ANALYSIS

On the other hand, to have a closer look at an outstanding performance of all algorithms, it is better to highlight each algorithm that provides superior performance on any problem with any dimension. Therefore, the minimum mean function error over 51 runs provided by an algorithm on the CEC 2017 functions with  $D = 10,30,50$  and  $100$  are summarized in Table 7. It must be noted that the empty cell means more than one algorithm provides the same mean function error. Besides, the number of cases in which each Algorithm provided minimum mean function error on the CEC 2017 functions with  $D = 10,30,50$  and  $100$  is shown in Figure 7.



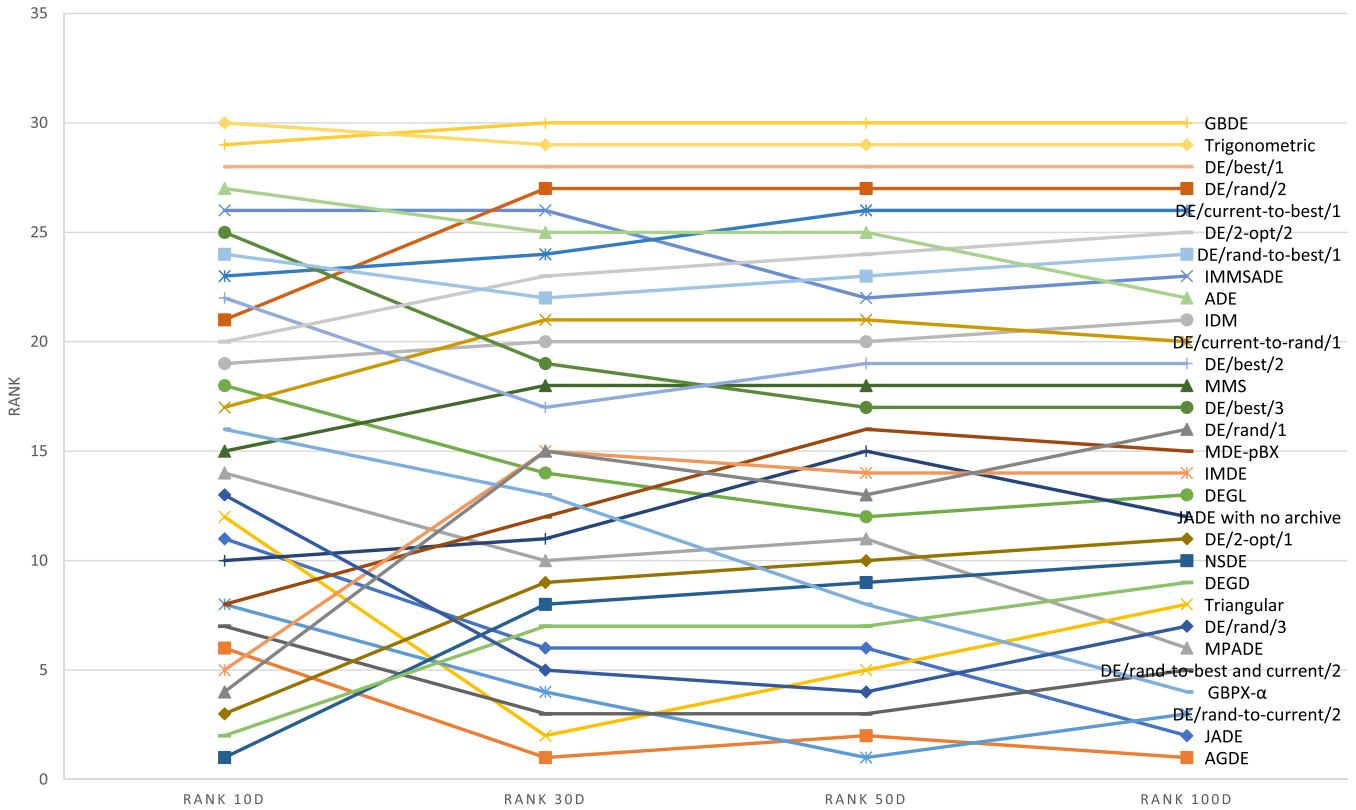


FIGURE 3. The rank of all algorithms on the CEC 2017 functions with  $D = 10, 30, 50$  and  $100$  according to function error.

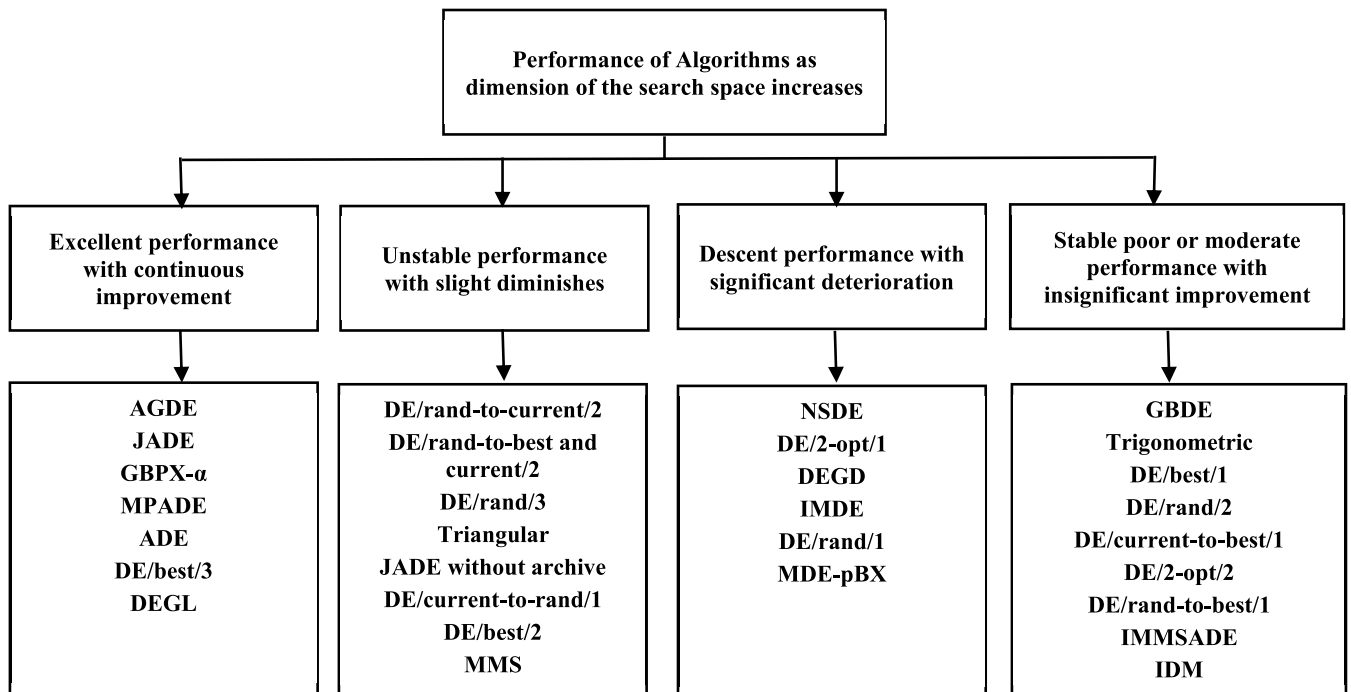


FIGURE 4. The classification of all algorithms according to their performances dimension of the space increases.

It can be clearly seen from Table 7 that 19 algorithms out of 30 algorithms can provide superior performance in solving at least one function with any dimension. Moreover, although no clear pattern can be observed, few algorithms provide

outstanding performance in many cases relative to others. Besides, it is clearly seen from Figure 7 that five, three and three algorithms provided minimum mean function error in one, two and three cases, respectively. Besides, AGDE,

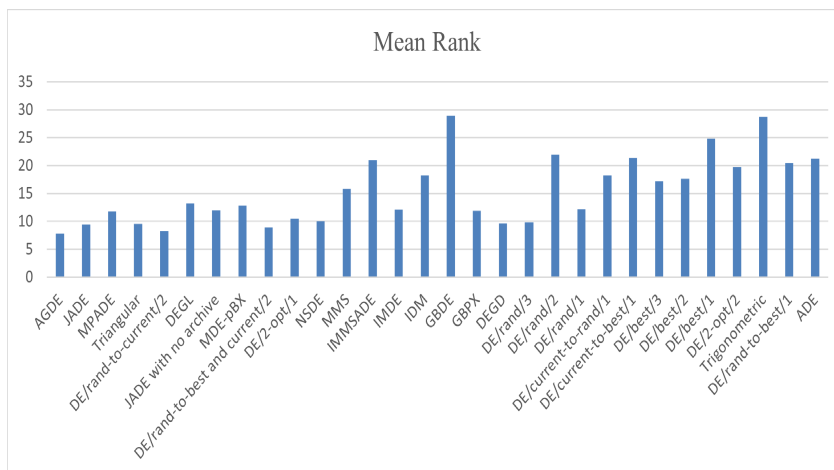


FIGURE 5. The mean rank of all algorithms on the CEC 2017 functions overall dimensions according to function error.

TABLE 3. Results of multiple-problem Wilcoxon’s test between AGDE and other algorithms for D = 10.

AGDE vs	R <sup>+</sup>	R <sup>-</sup>	p-value	+	≈	-	Dec.
JADE	133.5	142.5	0.891	12	6	11	≈
MPAAE	177	99	0.235	16	6	7	≈
Triangular	157.5	118.5	0.553	15	6	8	≈
DE/rand-to-current/2	145	131	0.831	15	6	8	≈
DEGL	218	131	0.280	18	3	8	≈
JADE with no archive	149	151	0.977	13	5	11	≈
MDE-pBX	132	144	0.855	11	6	12	≈
DE/rand-to-best and current/2	134.5	141.5	0.915	14	6	9	≈
DE/2-opt/1	81	195	0.083	8	6	15	≈
NSDE	61	192	0.033	10	7	12	-
MMS	183	195	0.885	17	2	10	≈
IMMSADE	340.5	94.5	0.083	24	0	5	≈
IMDE	100	200	0.153	9	5	15	≈
IDM	255.5	150.5	0.232	20	1	8	≈
GBDE	422	13	0.000	27	0	2	+
GBPX-α	198	180	0.829	16	2	11	+
DEGD	75	225	0.032	9	5	15	-
DE/rand/3	192	133	0.427	18	4	7	≈
DE/rand/2	240.5	110.5	0.099	20	3	6	≈
DE/rand/1	98.5	201.5	0.141	8	5	16	≈
DE/current-to-rand/1	236	142	0.259	20	2	7	≈
DE/current-to-best/1	309	97	0.016	20	8	1	+
DE/best/3	319	32	0.000	23	3	3	+
DE/best/2	296	55	0.002	22	3	4	+
DE/best/1	422	13	0.000	26	0	3	+
DE/2-opt/2	196	80	0.078	18	6	5	≈
Trigonometric	435	0	0.000	29	0	0	+
DE/rand-to-best/1	318	117	0.030	21	0	8	+
ADE	351	0	0.000	26	3	0	+

IMDE, and DE/ rand-to-current/2 get the first, second and third ranking as it can provide minimum mean function error in 16, 14 and 12 cases, respectively. Taking into consideration that regarding the algorithms that provided the largest number of cases with minimum mean function error in each dimension, GPBX, DE/2-opt/1 and DE/rand/1 provide the minimum mean function error in 4 cases in D = 10, AGDE provide superior performance in 7 and 6 cases in D = 30 and 50, respectively. In 100 dimensions, MPAAE provides the minimum mean function error in 7 cases. Nonetheless, it can be obviously deduced that AGDE provides more consistent

TABLE 4. Results of multiple-problem Wilcoxon’s test between AGDE and other algorithms for D = 30.

AGDE vs	R <sup>+</sup>	R <sup>-</sup>	p-value	+	≈	-	Dec.
JADE	273	133	0.111	19	1	9	≈
MPAAE	255	70	0.013	18	4	7	+
Triangular	204	147	0.469	17	3	9	≈
DE/rand-to-current/2	272	106	0.046	19	2	8	+
DEGL	320.5	85.5	0.007	22	1	6	+
JADE with no archive	291	115	0.045	20	1	8	+
MDE-pBX	305	73	0.005	22	2	5	+
DE/rand-to-best and current/2	189.5	135.5	0.468	16	4	9	≈
DE/2-opt/1	274.5	76.5	0.012	17	3	9	+
NSDE	270	108	0.052	21	2	6	≈
MMS	390	45	0.000	25	0	4	+
IMMSADE	400	35	0.000	27	0	2	+
IMDE	298	53	0.002	21	3	5	+
IDM	433	2	0.000	28	0	1	+
GBDE	435	0	0.000	29	0	0	+
GBPX-α	315	63	0.002	22	2	5	+
DEGD	276	102	0.037	20	2	7	+
DE/rand/3	267	111	0.061	18	2	9	≈
DE/rand/2	435	0	0.000	29	0	0	+
DE/rand/1	302	49	0.001	20	3	6	+
DE/current-to-rand/1	421	14	0.000	27	0	2	+
DE/current-to-best/1	407	28	0.000	27	0	2	+
DE/best/3	341	65	0.002	24	1	4	+
DE/best/2	343	35	0.000	24	2	3	+
DE/best/1	411	24	0.000	27	0	2	+
DE/2-opt/2	377	1	0.000	26	2	1	≈
Trigonometric	430	5	0.000	28	0	1	+
DE/rand-to-best/1	406	29	0.000	27	0	2	+
ADE	402	4	0.000	27	1	1	+

results and stable performance than MPAAE in D = 100 as AGDE gets the first ranking while MPAAE gets sixth ranking according to Friedman test.

### V. CORRECT CONVERGENCE VERSUS FALSE CONVERGENCE

In this section, the theoretical background, and the taxonomy of the convergence behavior of population-based-algorithm (general case) and DE algorithm (special case) are discussed. Besides, based on the results provided by all mutations, numerical experiments, and complete analysis of

**TABLE 5. Results of multiple-problem Wilcoxon’s test between AGDE and other algorithms for D = 50.**

AGDE vs	R <sup>+</sup>	R <sup>-</sup>	p-value	+	≈	-	Dec.
JADE	268	187	0.275	20	0	9	≈ ≈
MPADE	280	118	0.088	19	2	8	≈ ≈
Triangular	316	90	0.010	22	1	6	+
DE/rand-to-current/2	166	269	0.285	10	0	19	≈ ≈
DEGL	305	130	0.058	23	0	6	≈ ≈
JADE with no archive	307	128	0.053	23	0	6	≈ ≈
MDE-pBX	316	119	0.033	24	0	5	+
DE/rand-to-best and current/2	255	180	0.417	19	0	10	≈ ≈
DE/2-opt/1	282	153	0.163	18	0	11	≈ ≈
NSDE	311	95	0.014	24	1	4	+
MMS	361	74	0.002	26	0	3	+
IMMSADE	396	39	0.000	27	0	2	+
IMDE	297	138	0.086	19	0	10	≈ ≈
IDM	382	53	0.000	26	0	3	+
GBDE	435	0	0.000	29	0	0	+
GBPX-α	350.5	84.5	0.004	22	0	7	+
DEGD	243.5	191.5	0.574	20	0	9	≈ ≈
DE/rand/3	236.5	169.5	0.446	17	1	11	≈ ≈
DE/rand/2	418	17	0.000	28	0	1	+
DE/rand/1	293	142	0.103	19	0	10	≈ ≈
DE/current-to-rand/1	385	50	0.000	26	0	3	+
DE/current-to-best/1	371	64	0.001	26	0	3	+
DE/best/3	323	112	0.023	22	0	7	+
DE/best/2	379	56	0.000	26	0	3	+
DE/best/1	377	58	0.001	26	0	3	+
DE/2-opt/2	404	31	0.000	26	0	3	+
Trigonometric	430	5	0.000	28	0	1	+
DE/rand-to-best/1	372	63	0.001	26	0	3	+
ADE	411	24	0.000	28	0	1	+

**TABLE 6. Results of multiple-problem Wilcoxon’s test between AGDE and other algorithms for D = 100.**

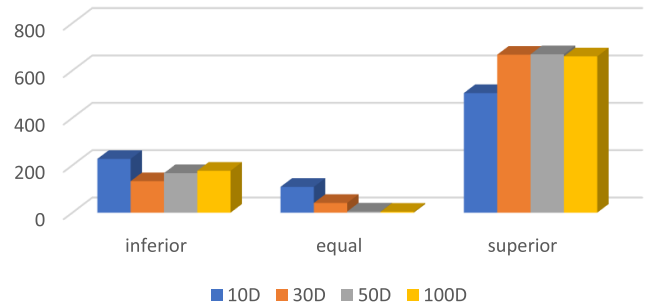
AGDE vs	R <sup>+</sup>	R <sup>-</sup>	p-value	+	≈	-	Dec.
JADE	194	241	0.611	16	0	13	≈ ≈
MPADE	198	237	0.673	14	0	15	≈ ≈
Triangular	344	91	0.006	24	0	5	+
DE/rand-to-current/2	220	215	0.957	14	0	15	≈ ≈
DEGL	255	180	0.417	21	0	8	≈ ≈
JADE with no archive	272	163	0.239	27	0	2	≈ ≈
MDE-pBX	279	156	0.184	22	0	7	≈ ≈
DE/rand-to-best and current/2	282	124	0.072	20	1	8	≈ ≈
DE/2-opt/1	276.5	129.5	0.094	19	1	9	≈ ≈
NSDE	327	108	0.018	24	0	5	+
MMS	297	138	0.086	23	0	6	≈ ≈
IMMSADE	370	85	0.001	25	0	4	+
IMDE	320	115	0.027	20	0	9	≈ ≈
IDM	391	44	0.000	26	0	3	+
GBDE	435	0	0.000	29	0	0	+
GBPX-α	271	164	0.247	15	0	14	≈ ≈
DEGD	302	133	0.068	23	0	6	≈ ≈
DE/rand/3	273	133	0.111	18	1	10	≈ ≈
DE/rand/2	414	21	0.000	28	0	1	+
DE/rand/1	330	105	0.015	21	0	8	+
DE/current-to-rand/1	396	39	0.000	26	0	3	+
DE/current-to-best/1	377	58	0.001	25	0	4	+
DE/best/3	248	158	0.305	20	1	8	≈ ≈
DE/best/2	380	55	0.000	27	0	2	+
DE/best/1	361	74	0.002	24	0	5	+
DE/2-opt/2	396	39	0.000	27	0	2	+
Trigonometric	432	3	0.000	28	0	1	+
DE/rand-to-best/1	370	85	0.001	25	0	4	+
ADE	435	0	0.000	29	0	0	+

the convergence behavior using two metrics, the diversity of population and success rate for all mutations are presented. Finally, the relationship between the quality of solution, diversity of population and success rate of mutation as is investigated.

**A. THEORETICAL BACKGROUND AND TAXONOMY OF THE CONVERGENCE BEHAVIOR**

In fact, Similar to all other Evolutionary algorithms (EAs), the evolutionary process of DE uses mutations, crossover, and

Statistical Results of AGDE



**FIGURE 6. Statistical comparison results of AGDE against other recent and state-of-the-art mutations with the growth of the dimensionality.**

selection operators at each generation to reach the global optimum. In simple DE, generally known as DE/rand/1/bin [2], an initial random population, denoted by  $P^{G=0}$ , consists of  $NP$  individual. Each individual is represented by the vector  $X_i = (x_{1i}, x_{2i}, \dots, x_{Di})$ , where  $D$  is the number of dimensions in solution space. Since the population will be varied with the running of the evolutionary process, the generation times in DE are expressed by  $G = 0, 1, 2, \dots, GEN$ , where  $G$  is the current generation and  $GEN$  is the maximal times of generations. For the  $i^{th}$  individual of  $P$  at the  $G$  generation, it is denoted by  $X_i^G = (x_{1i}^G, x_{2i}^G, \dots, x_{Di}^G)$ . The lower and upper bounds in each dimension of search space are respectively recorded by  $X_L = (x_{1L}, x_{2L}, \dots, x_{DL})$  and  $X_U = (x_{1U}, x_{2U}, \dots, x_{DU})$ . The initial population  $P^{G=0}$  is randomly generated according to a uniform distribution within the lower and upper boundaries  $(x_L, x_U)$ . After initialization, these individuals are evolved by DE operators (mutation and crossover) to generate a trial vector. A comparison between the parent and its trial vector is then done to select the vector which should survive to the next generation [23]. After that, during many generations of optimization process, the convergence of the population at last generation  $p^{G=Gmax}$  will be either correct or false. Correct convergence means that the DE algorithm has succeeded to reach the global optimal solution during the optimization process within a pre-determined or specified maximum number of generations i.e., the population successfully converges to the global optimal solution. On the contrary, false convergence means that the algorithm has failed to reach the global optimal solution during the optimization process within a pre-determined or specified maximum number of generations i.e., the population did not truly converge to the global optimal solution. In fact, the success of the population-based search algorithms is based on balancing two contradictory aspects: global exploration ability and local exploitation tendency [49]. Actually, the effectiveness and efficiency of any population-based algorithm is measured by its capability of producing high quality solution with high convergence rate, respectively. Moreover, the mutation scheme plays a vital role in DE search ability to produce high quality solution with high convergence rate. Virtually, the main cause of correct convergence is the ability

**TABLE 7.** The minimum mean function error over 51 runs provided by an algorithm on the CEC 2017 functions with  $D = 10,30,50$  and  $100$ , The empty cell means more than one algorithm provides the same mean function error.

F	D=10	D=30	D=50	D=100
1				
3				JADE
4		DE/best/3	JADE	MPADE
5	GPBX- $\alpha$	Triangular	DE/rand-to-current/2	DE/rand-to-current/2
6			IMDE	IMDE
7	GPBX- $\alpha$	GPBX- $\alpha$	GPBX- $\alpha$	DE/rand-to-current/2
8	GPBX- $\alpha$	Triangular	GPBX- $\alpha$	DE/rand-to-current/2
9			IMDE	IMDE
10	GPBX- $\alpha$	GPBX- $\alpha$	NSDE	NSDE
11	DE/rand/1	DE/rand-to-current/2	DE/rand-to-current/2	AGDE
12	IMDE	MPADE	DEGL	MPADE
13	DE/2-opt/1	Triangular	MPADE	MPADE
14	DE/rand/1	AGDE	AGDE	MPADE
15	IMDE	AGDE	AGDE	MPADE
16	DE/rand/1	AGDE	AGDE	AGDE
17	DE/2-opt/1	Triangular	AGDE	Triangular
18	DE/rand/1	AGDE	MPADE	JADE
19	DE/2-opt/1	AGDE	AGDE	MPADE
20	IMDE	DE/2-opt/1	AGDE	GPBX- $\alpha$
21	GBDE	Triangular	GPBX- $\alpha$	DE/rand-to-current/2
22	DE/rand/3	DE/2-opt/1	DEGD	DEGD
23	IDM	AGDE	DE/rand-to-current/2	DE/2-opt/1
24	IMMSADE	IDM	DE/rand-to-current/2	DE/rand-to-current/2
25	DE/rand/2	IMDE	IMDE	GPBX- $\alpha$
26	MMS	IMMSADE	DEGD	DE/2-opt/1
27	DE/rand/2	IMDE	IMDE	IMDE
28		IMDE	DE/rand/1	MPADE
29	DE/2-opt/1	AGDE	DE/rand-to-current/2	AGDE
30	DE/rand-to-current/2	MPADE	IMDE	IMDE

of the DE algorithm in balancing two contradictory aspects the global exploration capability and the local exploitation tendency. On the other hand, false convergence can be caused due to three different situations: (1) slow convergence (2) premature convergence and (3) stagnation. In these three situations, False convergence means that (1) the DE algorithm is unable to balance between both exploration capability and local tendency, (2) the DE algorithm favors considerably exploitation due to design of the mutations, (3) the DE

**TABLE 8.** The algorithmic design status in four convergence scenarios.

	Exploration capability	Exploitation tendency	Algorithmic design status
<b>Correct convergence</b>	Exist	Exist	(=) Balance both
<b>Slow convergence</b>	Exist	Exist	( $\approx$ ) Imbalance both
<b>Premature convergence</b>	Does not exist	Exist	( $\ll$ ) significant lack of exploration capability
<b>Stagnation</b>	Exist	Does not exist	( $\gg$ ) significant lack of exploitation tendency

( $\ll$ ): means that the algorithmic design favors exploitation or the exploration capability is much less than the exploitation tendency.

( $\gg$ ): means that the algorithmic design favors exploration or the exploitation tendency is much less than the exploration capability.

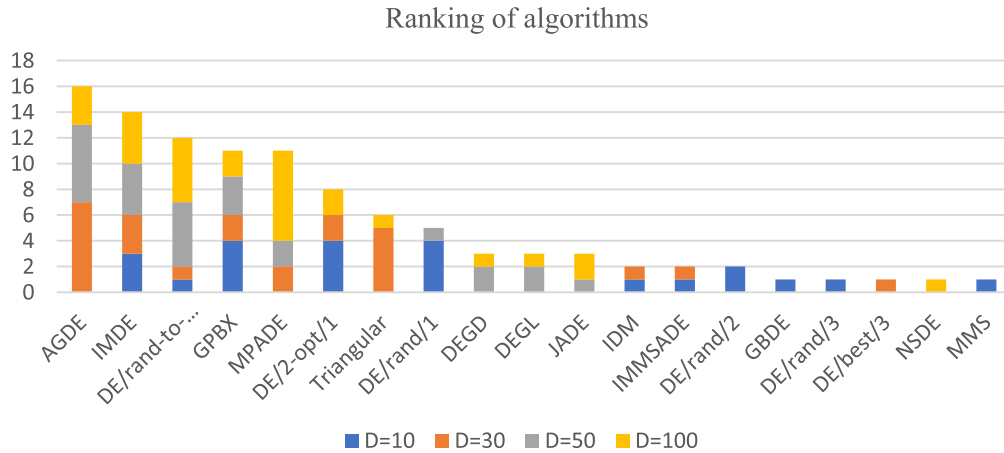
algorithm favors considerably exploration due to design of the mutations, during the search process, respectively. Thus, it must be noted that the exploration and exploitation abilities exist in the algorithmic design in case of slow convergence, but it is unable to balance both of them during the optimization process. However, the exploration capability may be very weak or does not exist in algorithmic design in case of premature convergence. Besides, regarding stagnation case, there is a lack of exploitation tendency or it may be does not exist in the algorithmic design of mutation. Table 8 describes these four convergence scenarios.

Actually, there are three possible reasons for false convergence such as (1) inappropriate configurations and/or improper adaptation schemes used to control the three main control parameters of DE (NP) population size, (F) Scaling factor and crossover rate (CR), (2) the mutation strategy may be either unable to balance both exploration and exploitation tendency or it favors one of both (exploration or exploitation) over another one due to its algorithmic design, or (3) both of them. Thus, the main objective of this section is to give a clear definition and theoretical explanation to describe these four scenarios of convergence behavior. Therefore, in order to accomplish this target, very important two metrics must be defined. Population diversity metric and success rate of mutation metric [39]. Besides, the mathematical definitions for both correct and false convergence must be given.

$$DP^G = \frac{1}{NP} \cot \sqrt{\sum_{i=1}^{NP} \left\| x_i^G - \frac{1}{NP} \cot \sum_{j=1}^{NP} x_j^G \right\|^2} \quad (66)$$

$$SR^G = \frac{NS^G}{NP} \quad (67)$$

where  $DP^G$  denotes the diversity of the population at the  $G$  generation.  $SR^G$  is the success rate of the population, and  $NS^G$  denotes the number of successful updates in generation  $G$ . Thus, the diversity of the population for each generation can



**FIGURE 7.** Number of cases in which each algorithm provided minimum mean function error on the CEC 2017 functions with D = 10, 30, 50 and 100.

be measured by the standard deviations of the individuals. Success rate can be measured by the number of trial vectors among the entire population that can successfully enter the next generation as they are better than their parents i.e., the number of trial vectors that survive to the next generation.

Besides, the correct convergence and false convergence can be expressed mathematically, respectively, as follows:

Correct convergence is given by

$$\lim_{G \rightarrow GEN} f(x) = f(x^*) \tag{68}$$

False convergence is given by

$$\lim_{G \rightarrow GEN} f(x) \neq f(x^*) \tag{69}$$

where G is the generation number, GEN is the maximum number of generations,  $f(x)$  denotes the objective function value of the best solution (x) obtained by algorithm in G generation, and  $f(x^*)$  is the objective function value of the well-known global optimum ( $x^*$ ) of each benchmark function. Thus, regarding correct convergence scenario, it is clearly seen from Eq. 68 that as G gets close to the GEN, the value of the function  $f(x)$  gets close to  $f(x^*)$ , which means  $f(x)$  converges to the true optimal solution. On the contrary, concerning false convergence, it is clearly seen from Eq. 69 that as G gets close to the GEN, the value of the function  $f(x)$  does not get close to  $f(x^*)$ , which means  $f(x)$  does not converge to the true optimal solution.

In order to fully understand and clearly differentiate between four scenarios, the diversity of the population  $DP^G$  and the success rate of the population  $SR^G$  must be analyzed for each scenario. Regarding correct convergence, it can be clearly deduced that from (68) that upon the best so far solution in G generation gets close to the global optimum solution, the remaining vectors will be attracted to the best solution and hence they will be clustered around it within few generations. Thus, the diversity of population DP will be gradually decreasing to 0, but SR is larger than 0 as there are many successful updates for trial vectors due to the attraction

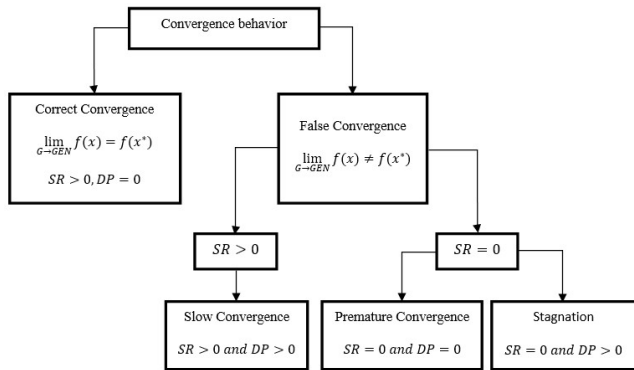
process. On the other hand, slow convergence means that the maximum number of generations has been reached while the population has not converged to a fixed point. Thus, there is neither clustering around the best so far solution nor attraction effect as there is no best solution found which means there is still slow continuous improvement for all population. Accordingly, the values of both metrics DP and SR greater than 0. Similar to slow convergence, stagnation means that the maximum number of generations has been reached while the population has not converged to a fixed point, but the population is unable to generate better solutions than current. Hence, the value of DP is greater than 0 while SR is 0. Premature convergence means that the population has converged to an inaccurate fixed point while the maximum number of generations has not been reached, Therefore, the diversity is completely lost within very few generations which implies that the values of both metrics DP and SR are 0. Actually, all vectors are clustered around the local best solution with zero distance and hence they have been quickly attracted to it within few generations and no more available updates. Using both DP and SR metrics, Table 9 presents the four scenarios and the classification of four convergence scenarios is further depicted in Figure 8. On other words, based on the values of DP and SR metrics, it can be deduced from Table 9 and Figure 8 that if SR is greater than 0 then the convergence behavior is based on the DP value. If DP is approximately zero, then it is the correct convergence scenario. Otherwise, it is the slow convergence scenario i.e., DP is greater than zero. On the other hand, if SR is approximately zero then the convergence behavior is based on the DP value. If DP is approximately zero, then it is the premature convergence scenario. Otherwise, it is the stagnation scenario i.e., DP is greater than zero.

Furthermore, to investigate the four scenarios of convergence based on DP and SR, the convergence behavior, DP and SR graphs (median curves) of AGDE algorithm on 10, 30 and 100-dimensional for some selected test functions are shown in Figure 9. Note that the median curves of AGDE algorithm



**TABLE 9. The four convergence scenarios with corresponding metrics values.**

	DP = 0	DP > 0
SR = 0	Premature convergence	Stagnation
SR > 0	Correct convergence	Slow convergence



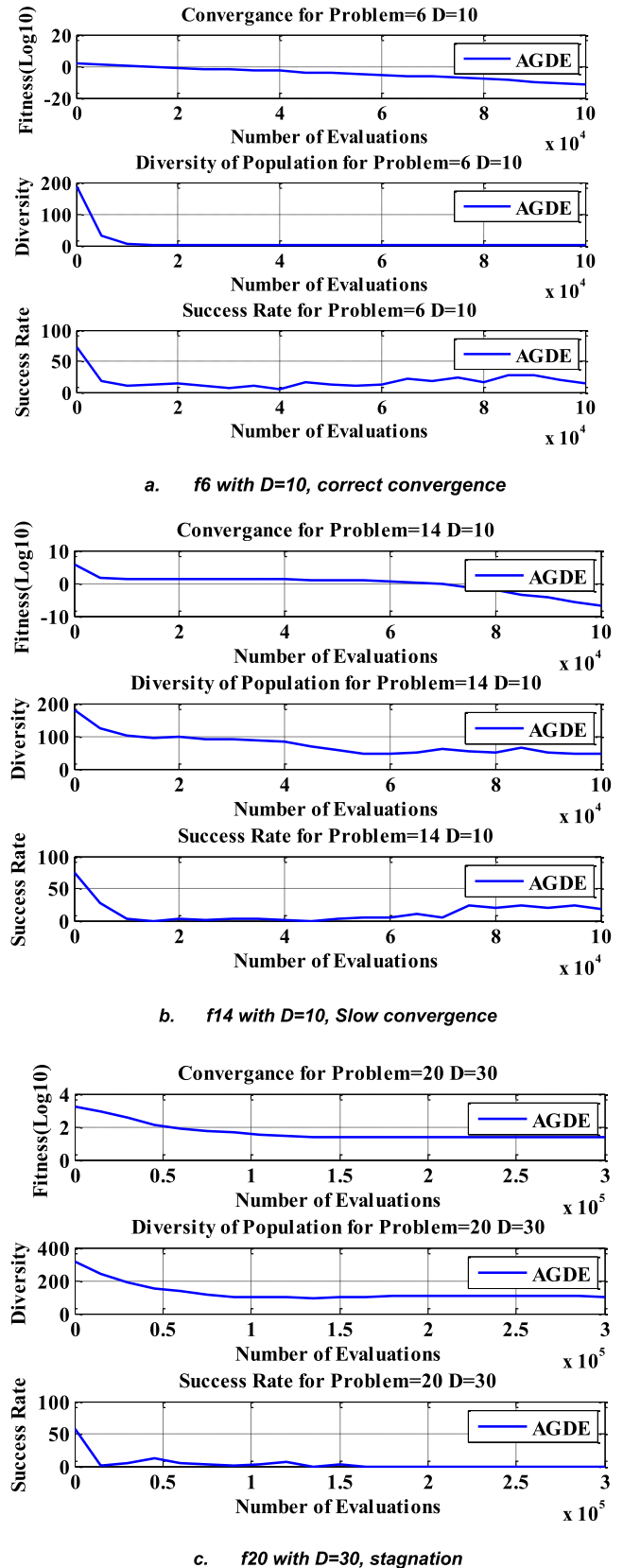
**FIGURE 8. The classification of four convergence scenarios using DP and SR metrics.**

on 10,30,50 and 100 dimensions for all test functions are illustrated in the supplemental file (Figure S2).

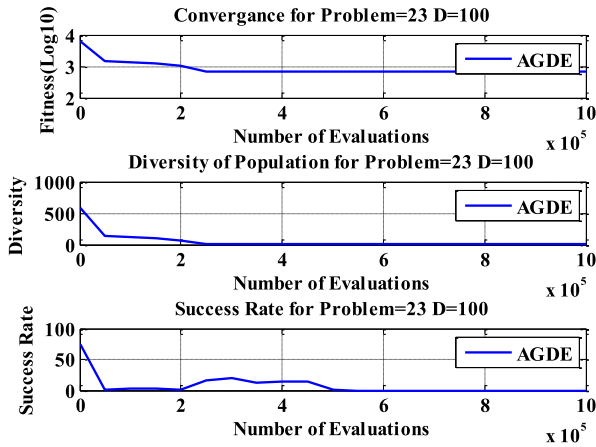
It can be seen from Figure 9 (a) that DP gradually decreases to 0, but SR is larger than 0. Thus, it is the correct convergence scenario. Second, it can be seen from Figure 9 (b) that DP and SR gradually decrease to a value that is larger than 0 steadily. This phenomenon is called slow convergence. Third, concerning stagnation scenario, it can be seen from Figure 9 (c) that DP gradually decreases to a value that is larger than 0, and then it remains unchanged. Besides, SR also decreases to close to 0, and then it remains unchanged as well. Finally, it can be obviously seen from Figure 9 (d) that both DP and SR reduce to 0 simultaneously. This is the premature convergence case.

**B. CONVERGENCE ANALYSIS FOR ALL ALGORITHMS**

In section IV, based on the solution quality provided by all algorithms, the performance of all mutations has been evaluated. Furthermore, it has been noted in previous sub-section V.A that the exploration capability of the mutation can be measured by evaluating the population diversity during the generations of the optimization process. Besides, the success rate SR measures the convergence ability of the mutation i.e., its ability of continuous improvement of trial vectors during the generations of the optimization process to reach a better solution. In this subsection, in order to analyze the superior and inferior performance of all algorithms, a complete analysis of the convergence behavior for all mutations are presented using aforementioned algorithmic design status and convergence scenarios classifications.



**FIGURE 9. Illustration of correct convergence, slow convergence, stagnation, and premature convergence.**



d. *f23 with D=100, Premature convergence*

**FIGURE 9. (Continued.)** Illustration of correct convergence, slow convergence stagnation, and premature convergence.

Firstly, Table 10 lists the average and final ranks for all algorithms across all problems and all dimensions according to success rate (SR). The best ranks are shown in bold and the second ranks are underlined. Besides, the final rank of all algorithms on the CEC 2017 functions is shown in Figure 10. It can be clearly seen from Table 10 that, regarding final ranking, IDM gets the first ranking followed by DE/current-to-rand/1 and DEGD while GBDE was the worst algorithm as it gets 30th ranking.

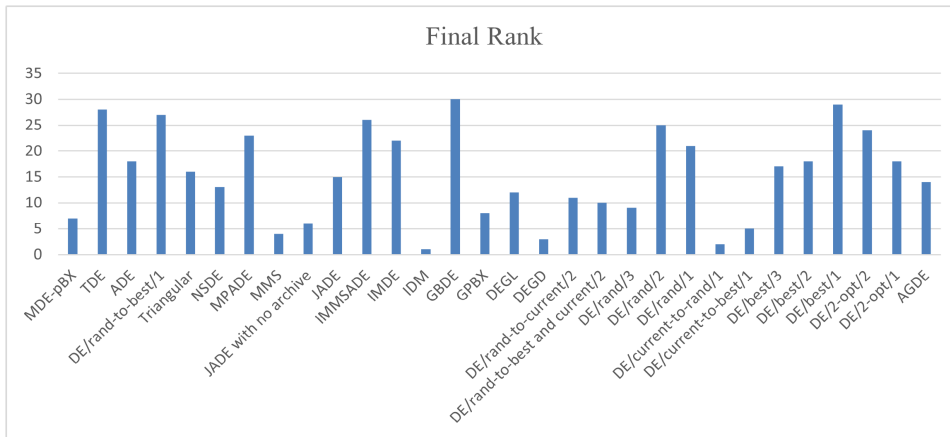
In order to analyze and compare all success rates (SR) provided by all algorithms, the ranking of the algorithms on the CEC 2017 functions with 10, 30, 50 and 100 dimensions are given in Figure 11. Firstly, similar to Figure 4 in subsection IV.C, it can be observed in Figure 11 that most of the algorithms perform differently on different dimensions. Generally, according to the growth of the search-space dimensionality from 10D to 100D, the success rate (SR) of all algorithms can be classified into four main categories as depicted in Figure 12.

The first category includes algorithms with excellent success rates (SR) that show outstanding continuous improvement in the value of their success rates (SR) as the dimension of the search-space increases i.e., it has a better rank in 100D than its rank in 10D with continuous improvement. The second category includes algorithms show instable success rates (SR) and slight diminishes in the value of their success rates (SR) as dimensions of the functions increases i.e., it has good rank in 10D with good performance, but its performance slightly fluctuates with slight deterioration with 30D, 50D, and 100D. The third category includes algorithms show descent success rates (SR) with complete and/or significant deterioration in the value of their success rates (SR) with the growth of the search-space dimensionality i.e., it has a better rank in 10D than its rank in 100D with continuous diminishes. The fourth category includes algorithms show stable moderate or poor success rates (SR) with insignificant improvement success rates (SR) with all dimensions and/or

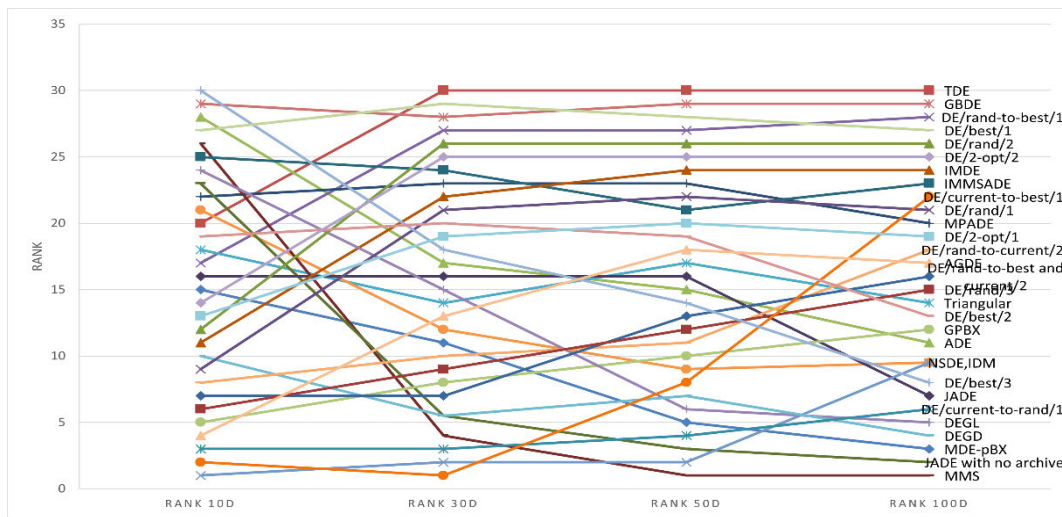
they get the lower ranking in most cases. It can be obviously shown from Figure 12 that MDE-pBX, ADE, NSDE, MMS, JADE with no archive, JADE, DEGL, DE/best/3, DE/best/2 mutations belong to the first category. The second category includes Triangular, IMMSADE, DEGD mutations. The third category contains DE/rand-to-best/1, IMDE, IDM, GPBX, DE/rand-to-current/2, DE/rand-to-best and current/2, DE/rand/3, DE/rand/2, DE/rand/1, DE/current-to-rand/1, DE/current-to-best/1, DE/2-opt/2, DE/2-opt/1, AGDE mutations. Finally, TDE, MPADE, GBDE, DE/best/1 mutations belong to the fourth category. Actually, it must be noted that not all algorithms that belong to a specific category show the same amount of improvement or deterioration in the value of their success rates (SR) with the growth of the search-space dimensionality. However, they show the same pattern. For instance, although DE/Current-to-rand/1 and DE/Current-to-best/1 belong to the third category, they show slight and considerable deteriorations respectively, in the value of their success rates (SR) such that DE/Current-to-rand/1 gets 3rd, 3rd, 4th and 6th ranking while DE/Current-to-best/1 2nd, 1st, 8th and 22<sup>nd</sup> ranking in 10D, 30D, 50D and 100D, respectively.

On the other hand, regarding the diversity of the population (DP), Table 11 lists the average and final ranks for all algorithms across all problems and all dimensions according to population diversity (DP). The best ranks are shown in bold and the second ranks are underlined. Besides, the final rank of all algorithms on the CEC 2017 functions is shown in Figure 13. It can be clearly seen from Table 11 that, regarding final rank, DE/rand/2 gets the first ranking followed by GBDE and DE/2-opt/2 while DE/rand-to-best/1 was the worst algorithm as it gets 30th ranking.

In order to analyze and compare all diversity of the population (DP) provided by all algorithms, the ranking of the algorithms on the CEC 2017 functions with 10, 30, 50 and 100 dimensions are given in Figure 14. Firstly, similar to Figure 11, it can be observed from Figure 14 that most of the algorithms perform differently on different dimensions. Generally, according to the growth of the search-space dimensionality from 10D to 100D, the diversity of the population (DP) of all algorithms can be classified into four main categories as depicted in Figure 15. The first category includes algorithms with an excellent diversity of the population (DP) that show outstanding continuous improvement in the value of their diversity of the population (DP) as the dimension of the search-space increases i.e., it has a better rank in 100D than its rank in 10D with continuous improvement. The second category includes algorithms show unstable diversity of the population (DP) and slight diminishes in the value of their diversity of the population (DP) as dimensions of the functions increases i.e., it has good rank in 10D with good performance, but its performance slightly fluctuates with slight deterioration with 30D, 50D, and 100D. The third category includes algorithms show descent diversity of the population (DP) with complete and/or significant deterioration in the value of their diversity of the population (DP) with



**FIGURE 10.** The final rank of all algorithms on the CEC 2017 functions overall dimensions according to success rate (SR).



**FIGURE 11.** The rank of all algorithms on the CEC 2017 functions with  $D = 10, 30, 50$  and  $100$  according to Success Rate (SR).

the growth of the search-space dimensionality i.e., it has a better rank in 10D than its rank in 100D with continuous diminishes. The fourth category includes algorithms show stable moderate or poor diversity of the population (DP) with insignificant improvement diversity of the population (DP) with all dimensions and/or they get the lower ranking in most cases.

It can be obviously shown from Figure 15 that triangular, NSDE, MMS, IMDE, GPBX, DEGD, DE/rand/1, DE/best/3, DE/best/2, DE/2-opt/2, DE/2-opt/1, DE/rand/2 mutations belong to the first category. The second category includes MPADE, JADE with no archive, AGDE mutations. the third category contains JADE, IMMSADE, IDM, GBDE, DE/rand-to-current/2, DE/ rand-to-best and current/2, DE/rand/3, DE/current-to-rand/1, DE/current-to-best/1 mutations. Finally, MDE-pBX, TDE, ADE, DE/rand-to-best/1, DEGL, DE/best/1 mutations belong to the fourth category. Actually, in the same context with success rate (SR),

it must be noted that not all algorithms that belong to a specific category show the same amount of improvement or deterioration in the value of their success rates (SR) with the growth of the search-space dimensionality. However, they show the same pattern. For instance, although GBDE and IDM belong to the third category, they show slight and considerable deteriorations respectively, in the value of their success rates (SR) such that DE/Current-to-rand/1 gets 1st, 2nd, 3rd and 3rd ranking while IDM 10th, 17th, 21st and 21st ranking in 10D, 30D, 50D and 100D, respectively.

In fact, it can be obviously seen from Figure 12 and Figure 15 that most of the algorithms belong to different categories according to the classifications of SR and DP Which are considered as empirical verification and practical validation to the theoretical contradictory aspects between both population diversity and convergence rate. For instance, JADE belongs to the first category according to the classification of (SR) while it belongs to the third category according

**TABLE 10.** Average ranks for all algorithms across all problems and all dimensions according to success rate SR.

Algorithm	10D	30D	50D	100D	Mean Rank	Final Rank
IDM	1	2	2	9.5	3.625	1
DE/current-to-rand/1	3	3	4	6	4	2
DEGD	10	5.5	7	4	6.625	3
MMS	26	4	1	1	8	4
DE/current-to-best/1	2	1	8	22	8.25	5
JADE with no archive	23	5.5	3	2	8.375	6
MDE-pBX	15	11	5	3	8.5	7
GPBX- $\alpha$	5	8	10	12	8.75	8
DE/rand/3	6	9	12	15	10.5	9
DE/rand-to-best and current/2	7	7	13	16	10.75	10
DE/rand-to-current/2	8	10	11	18	11.75	11
DEGL	24	15	6	5	12.5	12
NSDE	21	12	9	9.5	12.875	13
AGDE	4	13	18	17	13	14
JADE	16	16	16	7	13.75	15
Triangular	18	14	17	14	15.75	16
DE/best/3	30	18	14	8	17.5	17
DE/2-opt/1	13	19	20	19	17.75	18
ADE	28	17	15	11	17.75	18
DE/best/2	19	20	19	13	17.75	18
DE/rand/1	9	21	22	21	18.25	21
IMDE	11	22	24	24	20.25	22
MPADE	22	23	23	20	22	23
DE/2-opt/2	14	25	25	25	22.25	24
DE/rand/2	12	26	26	26	22.5	25
IMMSADE	25	24	21	23	23.25	26
DE/rand-to-best/1	17	27	27	28	24.75	27
TDE	20	30	30	30	27.5	28
DE/best/1	27	29	28	27	27.75	29
GBDE	29	28	29	29	28.75	30

to the classification of (DP). Thus, it means that JADE shows excellent success rate with continuous improvement while it shows Descent population diversity with significant deterioration as the dimension of the space increases. On the contrary, IMDE shows excellent population diversity with continuous improvement while it shows Descent success rate with significant deterioration as the dimension of the space increases.

Finally, in order to accomplish the target of this section, an empirical investigation of the four scenarios of the convergence behavior for all algorithms is applied. Therefore, using the error value, DP and SR values of the median run of all algorithms on 10, 30, 50 and 100-dimensional for all test functions, the four scenarios of convergence behavior are presented in Table 12. Furthermore, the percentage of cor-

**TABLE 11.** Average ranks for all algorithms across all problems and all dimensions according to the diversity of population DP.

	10D	30D	50D	100D	Mean Rank	Final Rank
DE/rand/2	2	1	1	1	1.25	1
GBDE	1	2	3	3	2.25	2
DE/2-opt/2	4	3	2	2	2.75	3
IMMSADE	3	4	4	6	4.25	4
IMDE	11	5	5	4	6.25	5
MPADE	5	7	10	8	7.5	6
DE/rand/1	13	6	6	5	7.5	6
DE/rand/3	7	8	8.5	9	8.125	8
ADE	6	9	8.5	10	8.375	9
DE/rand-to-current/2	9	10	11	12	10.5	10
DE/2-opt/1	17	11	7	7	10.5	10
DE/rand-to-best and current/2	12	12.5	14	15	13.375	12
DE/best/2	18	12.5	12	11	13.375	12
DE/current-to-rand/1	8	14	17	20	14.75	14
AGDE	14	18	16	14	15.5	15
DEGD	22	15	13	13	15.75	16
MDE-Pbx	16	16	15	18	16.25	17
IDM	10	17	21	21	17.25	18
GPBX- $\alpha$	19	19	18	16	18	19
NSDE	26	20	19	17	20.5	20
Triangular	25	21	20	19	21.25	21
DE/current-to-best/1	15	22	24	26	21.75	22
JADE with no archive	21	24.5	23	25	23.375	23
MMS	27	23	22	22	23.5	24
DEGL	24	24.5	25	24	24.375	25
JADE	20	26	27	27	25	26
DE/best/3	28	28	26	23	26.25	27
TDE	23	27	28	28	26.5	28
DE/best/1	30	29	29	29	29.25	29
DE/rand-to-best/1	29	30	30	30	29.75	30

rect (C), slow (SL), premature convergence (P) and stagnation (S) scenarios provided by all algorithms on 10, 30, 50 and 100-dimensional for all test functions are shown in Figure 16.

From Figure 16, it can be obviously seen that JADE, DE/rand-to-best and current/2, trigonometric and GBDE get the first ranking with the highest percentage of correct convergence, slow convergence, premature convergence, and stagnation, respectively. Besides, the percentage of four convergence scenarios behavior provided by all algorithms on the CEC 2017 functions overall dimensions is also presented.

Besides, it can be observed from Table 12 that all algorithms perform differently on different dimensions which means that the frequency of each scenario changes over the dimensions. Firstly, the percentage of the correct convergence of all algorithms is very low due to the difficulty of the benchmark test functions especially as the dimension of the

**TABLE 12.** The frequency and overall percentage of each convergence scenario provided by all algorithms on the CEC 2017 functions overall.

Algorithm	D=10				D=30				D=50				D=100				PERCENTAGE			
	C	SL	P	ST	C	SL	P	ST	C	SL	P	ST	C	SL	P	ST	C	SL	P	ST
JADE	7	7	9	6	2	9	13	5	2	11	13	3	1	18	8	2	10.34%	38.79%	37.07%	13.79%
AGDE	5	18	4	2	2	16	8	3	0	10	12	7	0	12	9	8	6.03%	48.28%	28.45%	17.24%
MPADE	3	8	5	13	2	1	6	20	1	3	5	20	1	5	3	20	6.03%	14.66%	16.38%	62.93%
Triangular	5	15	3	6	1	13	4	11	0	15	3	11	0	16	1	12	5.17%	50.86%	9.48%	34.48%
DE/rand-to-current/2	4	24	0	1	1	25	1	2	0	27	1	1	0	22	4	3	4.31%	84.48%	5.17%	6.03%
DEGL	5	6	13	5	2	11	10	6	1	15	9	4	0	19	7	3	6.90%	43.97%	33.62%	15.52%
JADE without Archive	7	8	10	4	2	16	9	2	0	19	9	1	0	20	7	2	7.76%	54.31%	30.17%	7.76%
MDE-pBX	7	11	7	4	2	13	8	6	0	19	4	6	0	20	4	5	7.76%	54.31%	19.83%	18.10%
DE/rand-to-best and current/2	4	24	1	0	2	23	2	2	0	25	4	0	0	21	6	2	5.17%	80.17%	11.21%	3.45%
DE/2-opt/1	4	9	9	7	2	6	6	15	0	8	5	16	0	8	6	15	5.17%	26.72%	22.41%	45.69%
DE/rand-to-best/1	2	2	24	1	0	0	29	0	0	0	29	0	0	1	29	1	1.72%	2.59%	95.69%	1.72%
DEGD	4	13	12	0	0	11	13	5	0	16	8	5	0	25	0	4	3.45%	56.03%	28.45%	12.07%
GBDE	0	0	0	29	0	0	0	29	0	0	0	29	0	0	0	29	0.00%	0.00%	0.00%	100.00%
GPBX- $\alpha$	0	16	12	1	1	17	9	2	1	16	9	3	0	13	9	7	1.72%	53.45%	33.62%	11.21%
IDM	0	24	1	4	0	20	0	9	0	22	0	7	0	16	9	4	0.00%	70.69%	8.62%	20.69%
IMDE	6	11	4	8	2	8	1	18	1	5	3	20	0	6	2	21	7.76%	25.86%	8.62%	57.76%
IMMSADE	0	4	0	25	0	1	0	28	0	0	0	29	0	0	0	29	0.00%	4.31%	0.00%	95.69%
MMS	4	7	16	2	1	16	11	1	0	21	8	0	0	22	7	0	4.31%	56.90%	36.21%	2.59%
NSDE	5	8	14	2	0	17	12	0	0	19	9	1	0	21	7	1	4.31%	56.03%	36.21%	3.45%
Trigonometric	0	0	29	0	0	0	29	0	0	0	29	0	0	0	29	0	0.00%	0.00%	100%	0.00%
ADE	5	0	3	21	1	5	3	20	0	11	1	17	0	11	3	15	5.17%	23.28%	8.62%	62.93%
DE/2-opt/2	4	8	0	17	0	8	0	21	0	6	0	23	0	1	0	28	3.45%	19.83%	0.00%	76.72%
DE/Best/1	0	3	26	0	0	1	28	0	1	1	27	0	0	2	27	0	0.86%	6.03%	93.10%	0.00%
DE/Best/2	3	5	14	7	2	5	6	16	1	7	3	18	0	13	4	12	5.17%	25.86%	23.28%	45.69%
DE/Best/3	3	3	21	2	2	8	14	5	1	12	15	1	0	19	8	2	5.17%	36.21%	50.00%	8.62%
DE/Current-to-Best/1	1	15	4	9	0	14	11	4	0	6	14	9	0	4	19	6	0.86%	33.62%	41.38%	24.14%
DE/Current-to-Rand/1	0	22	2	5	0	21	0	8	0	21	0	8	0	20	3	6	0.00%	72.41%	4.31%	23.28%
DE/Rand/1	4	13	5	7	1	11	3	14	1	5	3	20	0	9	1	19	5.17%	32.76%	10.34%	51.72%
DE/Rand/2	0	15	0	14	0	7	0	22	0	2	0	27	0	0	0	0	0.00%	20.69%	0.00%	54.31%
DE/Rand/3	1	22	0	6	0	21	0	8	0	21	1	7	0	20	2	7	0.86%	72.41%	2.59%	24.14%

problem increases. Secondly, for the remaining three scenarios, in order to identify the common scenario that is followed for each algorithm, the following rule based on the median values is applied. By excluding the percentage of the correct convergence scenario, if the percentage of one scenario is greater than the sum of the percentages for the other two scenarios or it represents at least 50% of total percentages of the three scenarios, this scenario is considered the most typical one for this algorithm i.e., it represents the trend of the search mechanism of this algorithm. Otherwise, this algorithm shows mixed convergence scenarios during optimization process with no obvious trend for a specific scenario. Consequently, based on the percentages of all scenarios provided by all algorithms, it can be clearly observed from Table 12 that the algorithms can be classified into four groups as depicted in Figure 17. The first group includes all algorithm that shows slow convergence behavior which are AGDE, Triangular, DE/rand-to-current/2, JADE with no archive, MDE-pBX, DE/ rand-to-best and current/2, DEGD, GPBX, IDM, MMS, NSDE, DE/current-to-rand/1 and DE/rand/3. The group of premature convergence behavior contains DE/rand-to-best/1, TDE, DE/best/1 and DE/best/3. Regarding the stagnation

behavior group, it includes MPADE, GBDE, IMDE, IMM-SADE, ADE, DE/2-opt/2, DE/rand/1 and DE/rand/2. Finally, JADE, DEGL, DE/2-opt/1, DE/best/2 and DE/current-to-best/1 belong to mixed convergence behavior group. In fact, regarding the first group, according to algorithmic design status in Table 8 and results in Table 12, all these mutations suffer from imbalance both global exploration ability and local exploitation tendency i.e., they are unable to reach the optimal solution within the maximum number of function evaluations (FEs). Nonetheless, the advantage of this group is the high success rate of improving the quality of solutions. Therefore, all these mutations are reliable and can be lonely used to design effective and efficient DE algorithm with satisfactory performance. However, further, improvement is a must to reach more accurate solutions to the global optimal solution. One possible suggestion is to adjust the parameters of mutation itself if exist. Another suggestion is to find manually or adaptively the optimal settings of the control parameters NP, CR and F to reach the basin of global optimal solution within the required maximum number of function evaluations (FEs). In the same context, the main drawback of the fourth group is that they show unstable pattern of convergence behavior



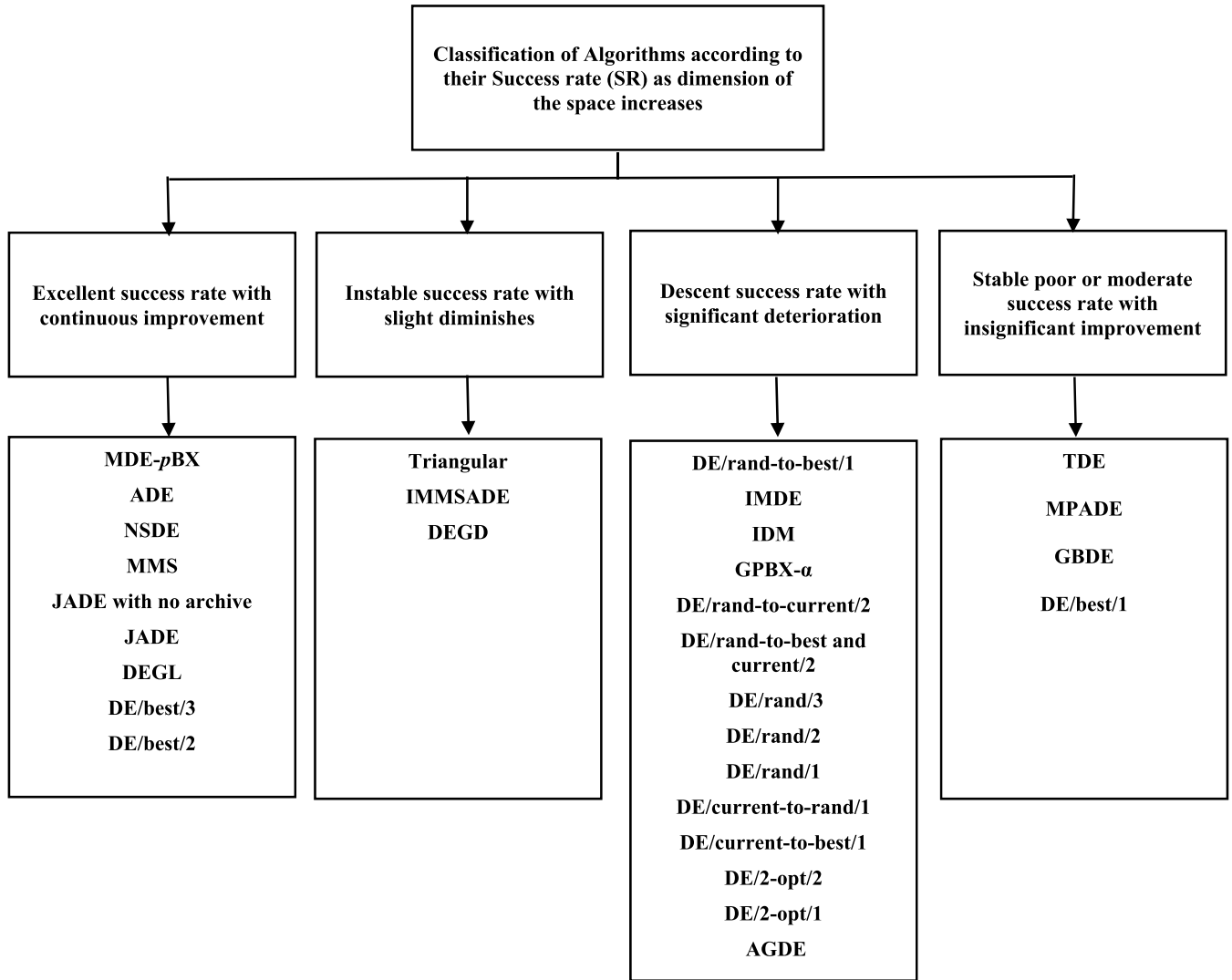


FIGURE 12. The classification of all algorithms according to their Success rate (SR) as the dimension of the space increases.

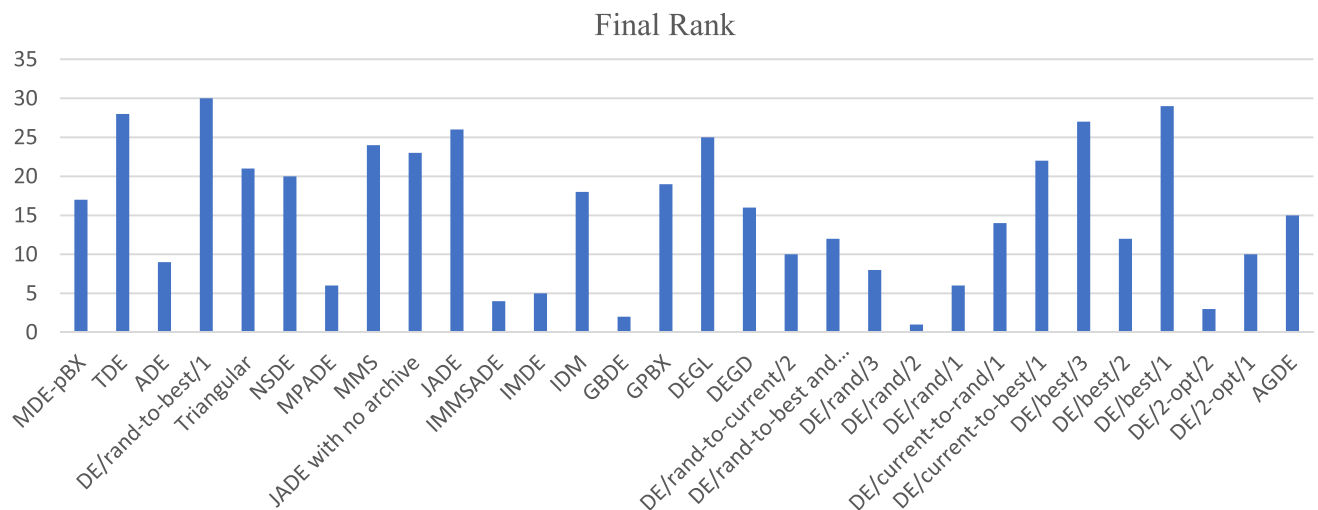


FIGURE 13. The mean rank of all algorithms on the CEC 2017 functions overall dimensions according to the diversity of population (DP).

although they still have a good tendency for slow convergence scenario. Therefore, the performance of these mutations can be easily further improved by following the same sugges-

tions for the first group which is improving the mutation and/or control parameters to at least increase its tendency for slow convergence scenario which in turn decreases the

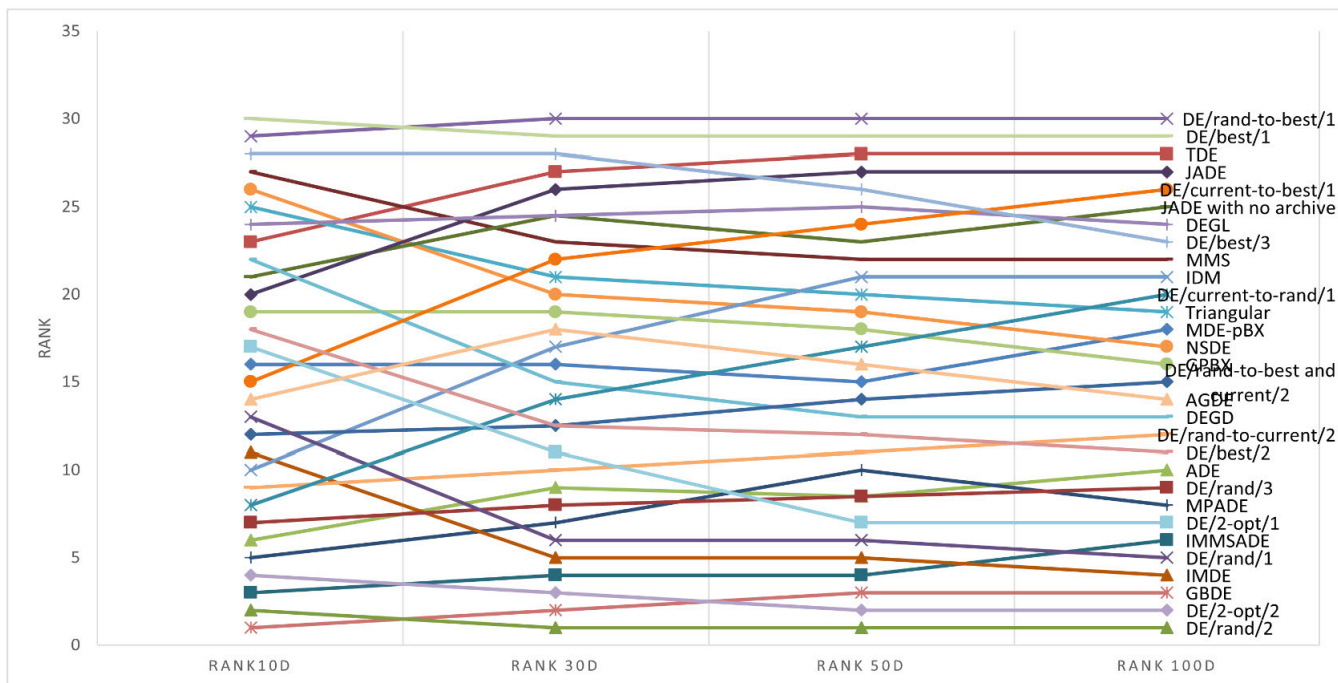


FIGURE 14. The rank of all algorithms on the CEC 2017 functions with  $D = 10, 30, 50$  and  $100$  according to the diversity of population (DP).

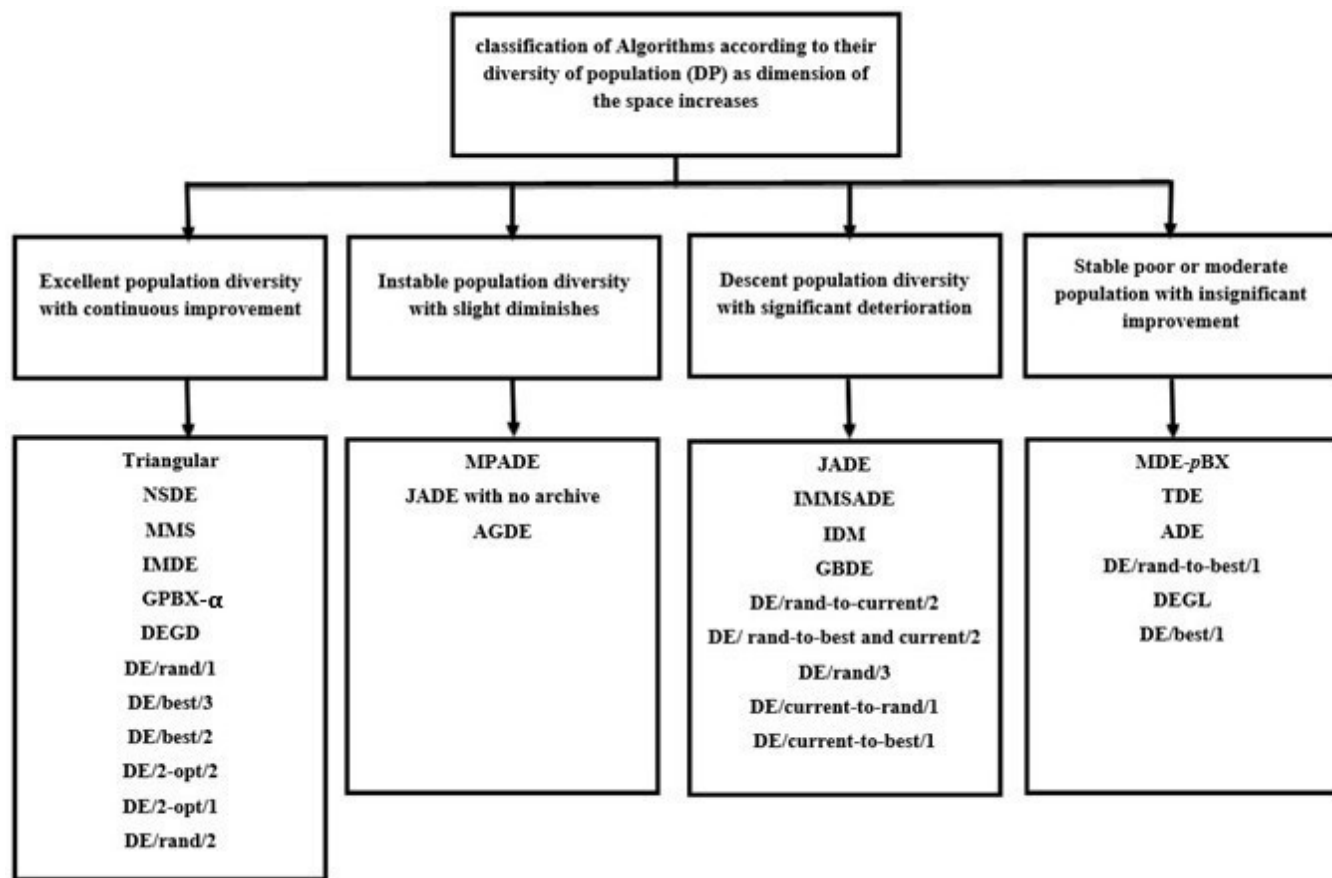


FIGURE 15. The classification of all algorithms according to their diversity of the population (DP) as the dimension of the space increases).

cases of premature convergence and stagnation. For instance, JADE algorithm shows mixed convergence scenarios due to switching off its adaptive schemes of CR and F. Therefore,

it can be deduced that the outstanding performance of JADE is mainly due to embedded adaptation mechanisms for CR and F parameters.

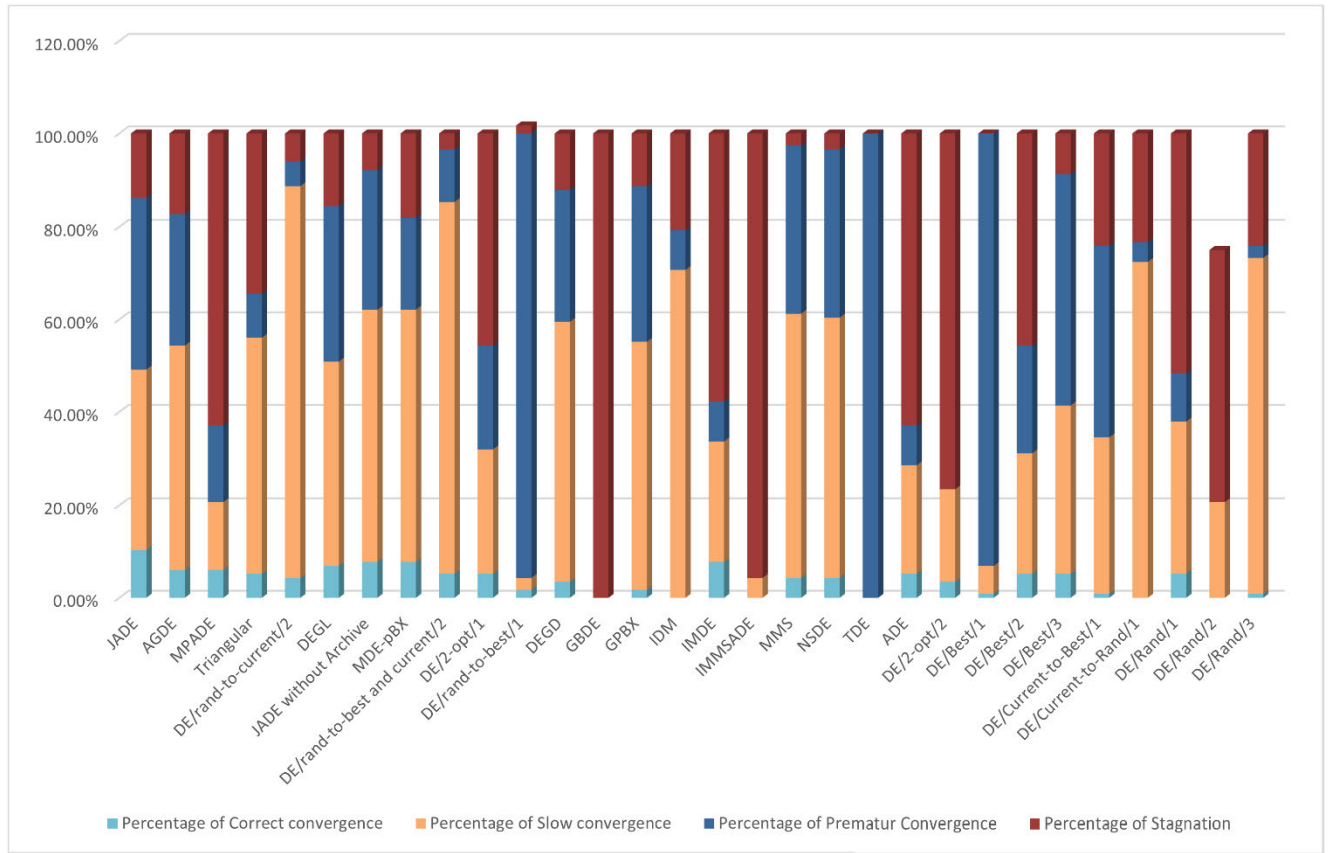


FIGURE 16. The percentage of four convergence scenarios behavior provided by all algorithms on the CEC 2017 functions overall dimensions.

On the other hand, concerning second and third group, according to Table 9, the success rate of these mutations in both situations is approximately zero. From Table 8, the premature convergence group has excessive exploitation tendency with very bad diversity while the stagnation group has excellent exploration ability with complete inability of improving solutions. Therefore, all these mutation strategies are not qualified enough to be lonely used in constructing an effective and efficient DE algorithm because their search mechanism must be repaired. The main idea behind repairing is that exploration ability must be added to premature convergence group and exploitation tendency must be added to stagnation group i.e., the search mechanism of each group can be balanced by adding its complement search ability. In fact, there are many possible suggestions to accomplish this target. Firstly, the mathematical expression of the mutation must be analyzed. For instance, it can be obviously seen from Table 12 and Figure 17 that the DE/best/1, DE/best/2 and DE/best/3 mutation strategies show fluctuate premature convergence behavior i.e., it is high, low and moderate, respectively. As a matter of fact, increasing difference vectors implies to higher diversity. Thus, it is expected that DE/best/2 and DE/best/3 mutation strategies show moderate and low premature convergence behavior, respectively. However, the effect of the best vector on the performance of these

mutations is higher than the effect of the difference vector(s). Thus, one possible suggestion to improve these mutations is to be mathematically formulated again as a linear combination between the best vector and the other difference terms and set the small value to the coefficient of the best vector and large value to the coefficient of difference vectors to reduce the excessive exploitative capability and increase the exploration ability. Furthermore, it can be clearly deduced from Table 12 and Figure 17 that the performance of DE/rand/1, DE/rand/2, and DE/rand/3 is considerably enhanced as the number of difference vectors increases and the base vector is random. Therefore, the second suggestion for repairing is by combining different mutation strategies with different features together to complement each other. Finally, to overcome the shortcoming of these mutations, hybridization with other evolutionary algorithms, local search operators is one alternative possible solution.

C. THE RELATIONSHIP BETWEEN THE THREE METRICS

In fact, in order to investigate the relationship between the three metrics diversity of the population, success rate and the quality of solution, the ranking of all algorithms according to the three metrics is shown in Figure 18. From Figure 18, as mentioned previously, it is obviously deduced that there

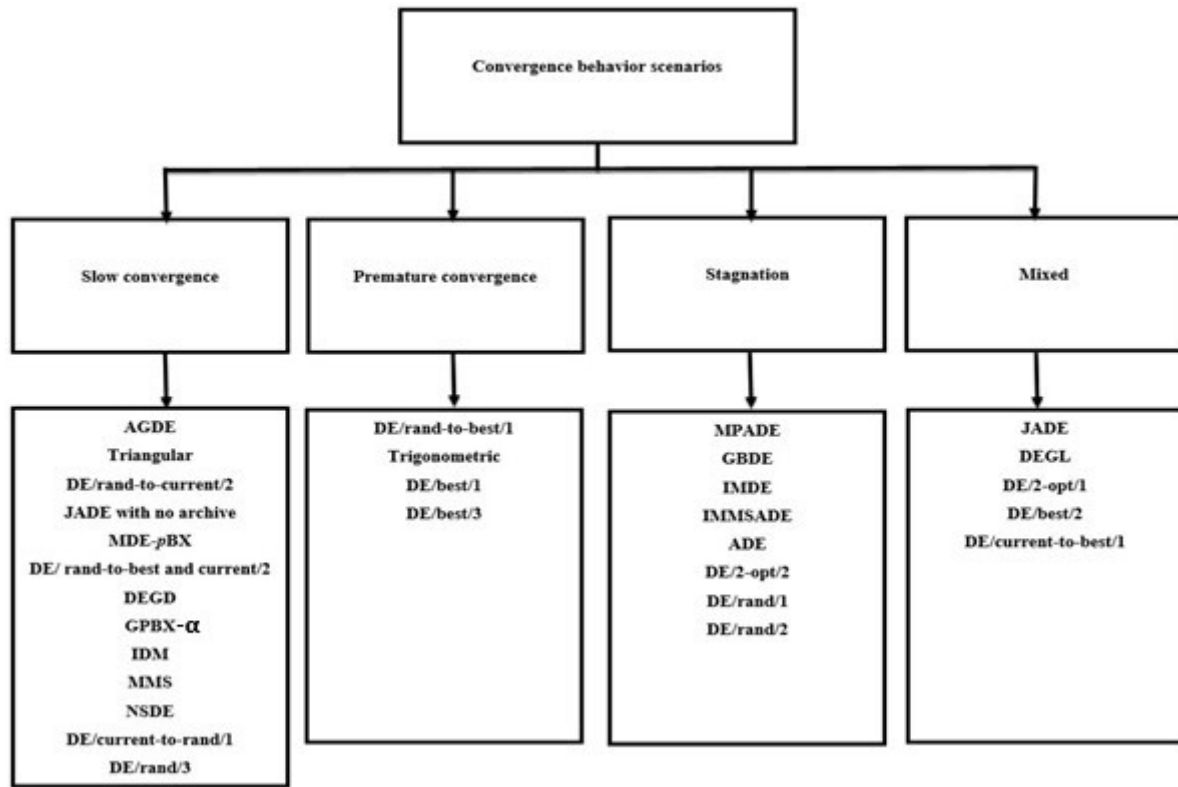


FIGURE 17. The classification of all algorithms according to their most typical convergence behavior scenario.

is contrary relationship between success rate and diversity of population such that they move in opposite directions for all algorithms with exception to DE/rand/3, DE/best/1, AGDE, DE/rand-to-current/2 and trigonometric mutations they get almost the same rank with both metrics. Besides, the performance of all DE-based algorithms is highly dependent on these two metrics. Thus, the quality of solution is significantly affected by population diversity and success rate. Accordingly, it would be greatly beneficial to derive the mathematical relationship between the quality of solution as the dependent variable and the diversity of population and success rate as independent variables.

Therefore, in order to derive the mathematical relationship between the quality of solution (QS) as dependent variable and the diversity of population (DP) and success rate (SR) as independent variables, the sum of median function errors (FES), the sum of median diversity of population and sum of median success rates across all problems of all algorithms in  $D = 10$  according to these three metrics were considered.

Three outlier values provided by the three algorithms IMMSADE, GBDE and trigonometric were removed. Curve fitting tool in MATLAB R2014a was used to fit a polynomial model of this relationship. Figures 19, 20, and 21 demonstrate the pattern of the scatter diagrams between the quality of solution (QS) as dependent variable and the diversity of population (DP) and success rate (SR) and between the diversity of population (DP) and success rate (SR), respectively. The rela-

tionship seems linear regarding (DP) and nonlinear regarding (SR). Besides, there is nonlinear relationship between (DP) and (SR). The generated model was:

$$\begin{aligned}
 QS = & a1 + a2 * DP + a3*SR + a4 * DP * SR \\
 & + a5*SR \wedge 2 + a6 * DP * SR \wedge 2 + a7 * SR \wedge 3 \\
 & + a8 * DP*SR \wedge 3 + a9 * SR \wedge 4
 \end{aligned}$$

Estimated values of coefficients  $a1-a9$  are shown in Table 13. The goodness of fit for this model was (R-square: 0.9677, Adjusted R-square: 0.9533, and RMSE: 665.5). This means that the estimated mathematical model is highly fitted to predict future data of quality of solution when the sum of the median values of (DP) and (SR) for  $D = 10$  is given. Note that  $D = 10$  is considered as a case study. Figure 22. Illustrates this model.

**VI. RECOMMENDATIONS, INSIGHTS, AND GUIDELINES**

Obviously, from the above results, comparisons and discussion, the following recommendations, insights, and guidelines must be taken into consideration when interested researchers would like to design effective and efficient DE algorithm. To be more precise and for a better explanation, recommendations, insights, and guidelines will be discussed form two directions: (1) quality of solution which is discussed in section 4, (2) convergence behavior which is discussed in section 5.

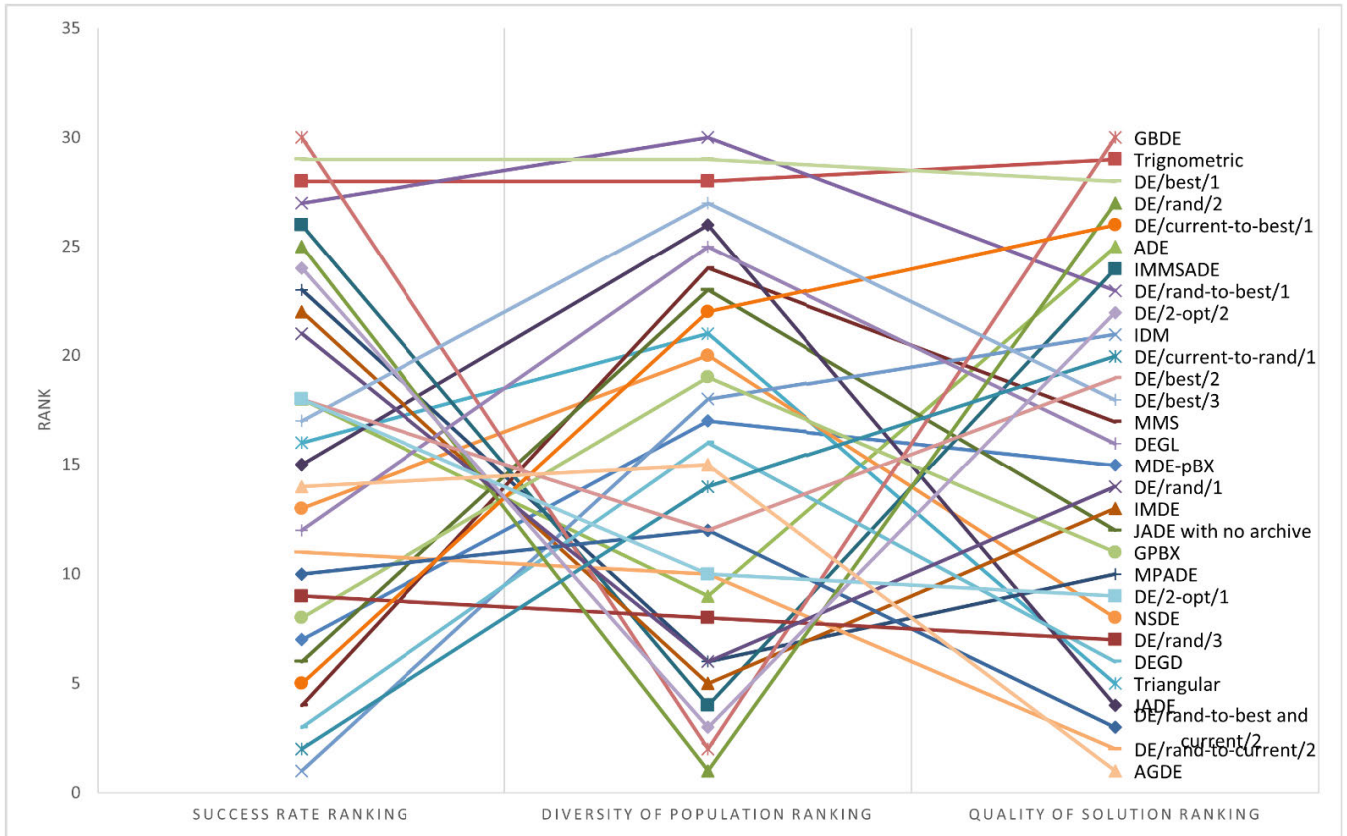


FIGURE 18. The ranking of all algorithms according to the three metrics success rate, diversity of population and quality of solution.

TABLE 13. Estimated values of coefficients  $a_1 - a_9$  for the model representing the relationship between QS, SR, and DP.

Coefficients	Estimated value
a1	3923
a2	-0.34
a3	1299
a4	-0.4653
a5	-1669
a6	1.667
a7	399.1
a8	-0.5101
a9	-8.33

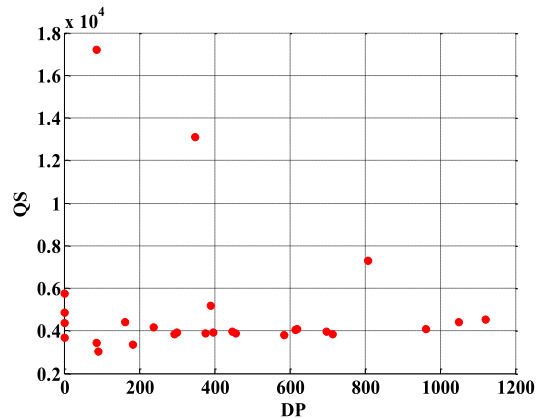


FIGURE 19. Scatter plot for the diversity of population (DP) and quality of solution (QS).

Firstly, regarding the **quality of solution**, which reflect the effectiveness of all evolutionary algorithms (in general) the DE-based algorithm (special case). The following recommendations, insights, and guidelines must be taken into consideration:

- 1) AGDE algorithm is of better searching quality, efficiency, and robustness for solving small, moderate, and high dimensions unconstrained global optimization problems. It is clear that AGDE algorithm per-

forms well and it has shown its outstanding superiority with separable, non-separable, unimodal, multimodal, hybrid and composition functions with shifts in dimensionality, rotation, multiplicative noise in fitness and composition of functions. Consequently, its performance is not influenced by all these obstacles. Contrarily, it greatly keeps the balance the local optimization speed and the global optimization diversity



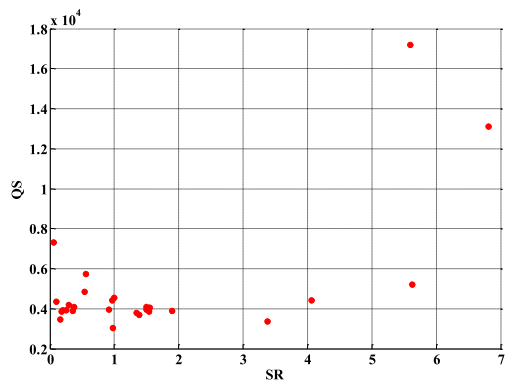


FIGURE 20. Scatter plot for success rate (SR) and quality of solution (QS).

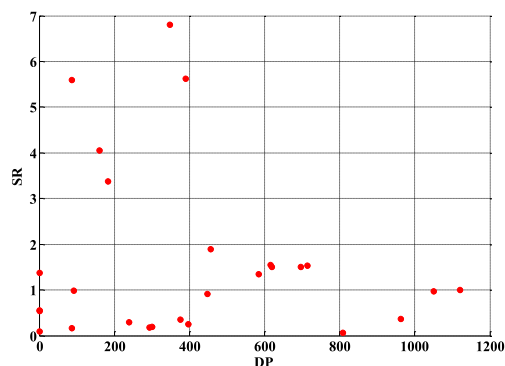


FIGURE 21. Scatter plot for the diversity of population (DP) and success rate (SR).

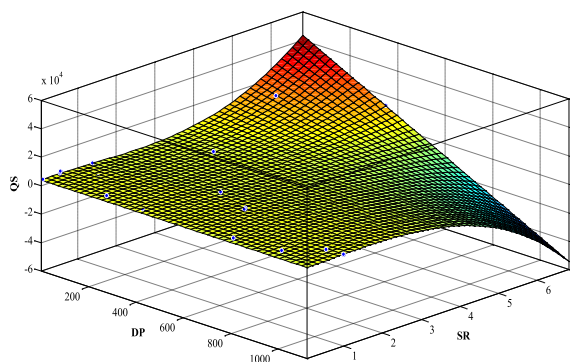


FIGURE 22. The fitted polynomial surface of the three metrics quality of solution (QS), diversity of population (DP) and success rate (SR).

in challenging optimization environment with invariant performance. Besides, its performance is superior and competitive with the performance of the recent and state-of-the-art well-known algorithms.

- 2) Obviously, the outstanding performance and great success of AGDE is due to the utilization of the incorporation of the objective function value in the mutation scheme. To be more precise, the fully directed perturbations in the proposed mutation of AGDE resembles the concept of the gradient as the difference vector is directed from the worst vectors to the best vectors.

Thus, it is considerably used to explore the landscape of the objective function being optimized in different sub-region around the best vectors within search space through optimization process [52].

- 3) Consequently, it can be deduced that by utilizing and sharing the best and worst information of the DE population, the proposed fully directed mutation of AGDE balances both global exploration capability and local exploitation tendency. However, it must be noted that AGDE, triangular, IDM, ADE and trigonometric get 1st, 5th, 21st, 25th and 29th ranking according to Friedman test. Therefore, although all these algorithms belong to fully directed mutation category, AGDE and triangular algorithms perform well and show outstanding and competitive performance compared with other algorithms in different categories. However, the remaining algorithms IDM, ADE, and trigonometric show poor performance on the majority of functions. Thus, it is not guaranteed that all fully directed mutations will show superior performance. It depends on the design of the mutation itself. By the way, it is noteworthy to mention that the idea of directed mutation scheme was firstly proposed by Mohamed, Sabry and Khorshid [49] as a novel contribution in designing ADE algorithm.
- 4) DE/rand-to-current/2 and DE/rand-to-best and current/2 get the second and third ranking although they belong to random and directed mutations groups, respectively. Furthermore, they maintain effectively the balance between the global exploration and local exploitation abilities during the search process and they outperform JADE in  $D = 10, 30$  and  $50$ . Actually, it must be noted that the design of these two mutations is better than another basic, classical and state-of-the-art mutations i.e. mutations with standard DE/X/Y/Z notation where DE denotes differential evolution, X denotes the target or base vector, Y denotes the number of randomly selected difference vectors and Z indicates type of crossover operator [23]. Really, these two mutations are highly recommended for further improvement.
- 5) In general, although JADE shows good performance with continuous improvement as the dimension of the functions increases, it is ranked fourth. Moreover, it seems that JADE is significantly deteriorated without archive as it gets 12th ranking which proves that the concept of the archive has the main effect on the performance of mutation of JADE. Accordingly, this concept must be widely used, and it must be enhanced.
- 6) Generally, the ranking of top ten algorithms are as follows: AGDE, DE/rand-to-current/2, DE/rand-to-best and current/2, JADE, triangular, DEGD, DE/rand/3, NSDE, DE/2-opt/1, and MPAGE. It can be clearly seen that 7 algorithms out of 10 belong to directed mutation group while the others 3 algorithms belong to random group algorithm. Besides, out of seven directed

- algorithms, 2 algorithms (AGDE and Triangular) are fully directed while the other five algorithms (DE/rand-to-best and current/2, JADE, NSDE, DE/2-opt/1, and MPAD) belong to partially directed group. Thus, it implies that the incorporation of the objective function value in the design of mutation scheme is better than pure randomness although both must be improved.
- 7) Regarding the remaining mutations in the random group that are not included in top ten ranked mutations, it is not recommended to be used alone in designing DE algorithm because they are unable to maintain the balance between the population diversity and convergence speed. Some of them keep diversity but with slow convergence speed i.e., it favors exploration while the others are exploitative mutations i.e., it loses diversity but with fast convergence speed. Thus, it must be hybrid with one or more top ten mutations.
  - 8) Concerning probabilistic group, it includes just one mutation, GBDE with the poorest performance as it gets the last ranking. Accordingly, it is highly recommended to pay more attention and much research effort is needed to repair this mutation first and then to improve this research direction.
  - 9) Generally speaking, refereeing to the empirical taxonomy in sub-section IV.C.1, it is preferred to improve the mutations in the second category, which shows unstable performance as dimension increases, by combing them with the mutations from the first category which shows outstanding improvement as dimension of the search-space increases. Thus, it is recommended to develop new hybridization using these two families of mutations.
  - 10) Finally, all mutations belong to third and fourth categories which show significant deterioration as dimension of the search-space increases or with all dimensions, they need further investigation for possible repair. In fact, most of these mutations suffer from premature convergence and/or stagnation in addition to the curse of dimensionality i.e., their performance is significantly deteriorated as the dimension of the search space increases. Besides, on the other hand, all mutations belong to first and second categories show excellent and satisfactory performance as the dimension of the search space increases. However, not all these mutations show the same convergence behavior i.e., they may have different merits and may suffer from different weakness. Therefore, it is suggested that the diversity of population and success rate and convergence speed of these mutations must be analyzed to proposing a novel idea for repairing for the first case or suggest an appropriate modification or improvements for the second case.

Secondly, regarding the **Convergence behavior**, which reflects the efficiency of all evolutionary algorithm (in general) and the DE-based algorithm (special case). The follow-

ing recommendations, insights, and guidelines must be taken into consideration:

- 1) In fact, the success of the population-based search algorithms is based on balancing two contradictory aspects: global exploration ability and local exploitation tendency [49]. Actually, the effectiveness and efficiency of any population-based algorithm are measured by its capability of producing a high-quality solution with high convergence rate, respectively. Moreover, the mutation scheme plays a vital role in DE search ability to produce a high-quality solution with high convergence rate.
- 2) However, until now, correct convergence scenario can be theoretically described but practically it cannot be consistently achieved over all runs and all dimensions with all benchmark functions due to its difficulty especially as the dimension of the search-space increases. Thus, slow convergence scenario is virtually considered as the most appropriate objective when designing DE-based algorithm.
- 3) Besides complete avoiding of the possible occurrence of both premature convergence and stagnation scenarios when solving benchmark functions with different dimensions or even real-world problem or application is considered impossible or at least it is a very rare event. However, minimizing its frequent occurrence is a must.
- 4) The diversity of population and success rate are two contradictory aspects. Therefore, the eminent performance of any evolutionary algorithm is based on balancing both of them during the optimization process. Thus, it is impossible to find one algorithm get 1st ranking in both metrics. From Table 10 and Table 11, IDM get 1st in success rate metric, while DE/rand/2 gets 1st in the diversity of population metric, respectively. On the contrary, it is possible to find one algorithm get the same ranking in both metrics especially when it shows very weak performance such as DE/best/1 get 29th ranking and trigonometric algorithms get 28th ranking in both metrics, respectively, from Tables X and XI. Besides, they get 28th ranking and 29th ranking in quality of solution ranking according to Table 2, respectively.
- 5) Besides, it is not guaranteed for any algorithm that being 1st in one of these aspects will greatly improve the performance as it may significantly deteriorate the other aspect. For instance, regarding the quality of solutions, the top four algorithms are AGDE, DE/rand-to-current/2 and DE/rand-to-best and current/2 and JADE, respectively. Besides, IDM and DE/rand/2 get 21st and 27th ranking, respectively. Regarding diversity of population and success rate, from Table 10 and Table 11, it can be observed that none of top four algorithms get first ranking in the diversity of population or success rate. AGDE, DE/rand-to-current/2 and DE/rand-to-best

and current/2 and JADE get 14th, 11th, 10th, and 15th ranking in success rate (SR) metric and they get 15th, 10th, 12th and 26th ranking in the diversity of population (DP) metric. On the contrary, IDM and DE/rand/2 get 1st and 25th ranking in success rate metric, respectively, while they get 18th and 1st in diversity metric, respectively.

- 6) Accordingly, Like the classification of all algorithms according to their performances (the solution quality provided) as the dimension of the space increases, there are two classifications of all algorithms according to their success rates and population diversity as the dimension of the search space increases. The position of most algorithms is not the same in all classifications. For instance, JADE belongs to the first category according to the classification of (SR) while it belongs to the third category according to the classification of (DP). Thus, it means that JADE shows excellent success rate with continuous improvement while it shows Descent population diversity with significant deterioration as the dimension of the space increases.
- 7) However, in above mentioned three classifications of mutations according to their performance success rate and diversity, it must be taken into consideration that not all mutations that belong to a specific category show the same pattern or the amount of improvement or deterioration with the growth of the search-space dimensionality. For instance, according to success rate classification, although GBDE and IDM belong to the third category, they show slight and considerable deteriorations respectively, in the value of their success rates (SR) such that GBDE gets 1st, 2nd, 3rd and 3rd ranking while IDM 10th, 17th, 21st and 21st ranking in 10D, 30D, 50D, and 100D, respectively.
- 8) All mutations belong to slow convergence scenario and mixed convergence scenario categories are reliable and can be lonely used to design effective and efficient DE algorithm with satisfactory performance. However, further, improvement is a must to reach more accurate solutions or very near solution to the global optimal solution. Therefore, the performance of these mutations can be easily further improved by finding manually or adaptively the optimal settings of the mutation parameter and/or control parameters NP, CR, F.
- 9) On the other hand, all mutation strategies belong to premature convergence scenario and stagnation scenario categories are not qualified enough to be lonely used in constructing an effective and efficient DE algorithm because their search mechanism must be repaired. There are three directions for possible repair such that (1) the mathematical expression of the mutation must be analyzed, (2) combining different mutation strategies with different features together to complement each other must be empirically investigated, (3) hybridization with other evolutionary algorithms,

local search operators are highly recommended as one alternative possible solution.

- 10) Besides, according to the convergence behavior analysis, all mutations belong to second and third categories (Premature convergence and stagnation), need further investigation for possible repair. Besides, all mutations belong to first and fourth categories (slow convergence and mixed convergence scenarios), need further improvement but they can still be used solely without involving other mutations to perform the optimization process. However, taken into consideration that not all algorithms that belong to a specific category provided similar quality of solutions i.e., for any two algorithms A and B, they may get the same percentages of convergence scenarios but one of them may produce better solution quality than the other. In other words, it may prematurely converge, or it can be stagnated very near to the global optimal solution. Alternatively, one of them may be slowly converged to a better solution than the other. Thus, the quality of solution provided by many algorithms that show the same convergence scenario must be checked to consider which algorithm is better than the other.

**Altogether**, it is recommended that the two contradictory aspects the **diversity of population (DP) and the convergence rate (SR)** must be taken into consideration in evaluating and comparing two or more algorithms in addition to the solution quality to perform a complete assessment.

## VII. CONCLUSION

This study represents a significant step and a considerable trend to outline the progress of existing different mutation schemes of DE algorithm that have been developed to solve global optimization problems. In this paper, comprehensive review of 30 DE novel mutations and 6 DE novel concepts that were proposed between 1995 and 2020 is proposed. This review is based on new theoretical taxonomy. The proposed taxonomy classifies all novel contributions of DE mutations into two main areas (1) Novel Mutation and (2) Novel concept. In the first area, an innovative mutation scheme is added to basic DE mutation. However, regarding the second area, a new technique or method is proposed to enhance the selection process of the individuals to form the mutation scheme. Then, based on the structure of the novel mutation, it can be further classified into three groups: (a) Random, (b) directed, and (c) probabilistic. Then, the directed mutation can be classified into two main categories: (1) partial directed and (2) full directed. Actually, it is noteworthy to mention that this is the first research paper that proposes a new taxonomy to classify the contributions of DE mutations.

The performance of the proposed DE-based algorithms using these mutations was tested on 29 benchmark functions proposed in the CEC 2017 special session on real-parameter optimization. Then, based on an overall performance and

comparison between all mutations over all dimensions, an empirical taxonomy is also provided. This taxonomy classifies all algorithms into four categories based on the performance of algorithms with the growth of the search-space dimensionality from 10D to 100D. The first category includes algorithms with excellent performance that show outstanding improvement as the dimension of the search-space increases. The second category includes algorithms show slight diminishes and instable performance as dimensions of the functions increases. The third category includes algorithms show complete and/or significant deterioration with the growth of the search-space dimensionality. The fourth category includes algorithms show almost the same moderate to poor performance with insignificant improvement in all dimensions. Furthermore, in order to statistically analyze the performances of all algorithms, two non-parametric tests (the Friedman test and Wilcoxon's test) are used with the significance level of 0.05.

As a summary of results, the performances of the AGDE algorithm were statistically superior to and competitive with other recent and well-known state-of-the-art algorithms in the majority of functions and for different dimensions. Furthermore, DE/rand-to-current/2 and DE/rand-to-best and current/2, JADE and triangular get 2nd, 3rd, 4th, and 5th places, respectively, and they show outstanding performance on the majority of functions. Besides, DEGD, DE/rand/3, NSDE, DE/2-opt/1 and MPADE get 6th, 7th, 8th, 9th, and 10th places, respectively, with promising solutions and competitive performance. Furthermore, following to theoretical taxonomy, 7 algorithms out of 10 belong to directed mutation group while the others 3 algorithms belong to random group algorithm which confirms that incorporation of objective function value in designing mutation scheme is better than pure randomness.

On the other hand, according to empirical taxonomy, AGDE, JADE, GBPX, and MPADE show perfect performance with continuous improvement as the dimension of the functions increases while NSDE, DE/2-opt/1, DEGD, IMDE, DE/rand/1 and MDE-pBX algorithms show complete deterioration as dimension increases. Besides, the convergence behavior of top ten algorithms is also analyzed. Then, the superior performance of all algorithms is presented. On the other hand, the theoretical background, and the taxonomy of the convergence behavior of population-based algorithm (general case) and DE algorithm (special case) are discussed. Besides, based on the results provided by all mutations, numerical experiments, and complete analysis of the convergence behavior for all mutations are presented using novel algorithmic design status and suggested convergence scenarios classifications which are based on the success rate and population diversity metrics.

Finally, recommendations, guidelines, insights, and suggestions for experienced practitioners and interested researchers in designing and developing effective and efficient DE algorithms to address various optimization problems in different fields are discussed.

Overall, based on results, tests, comparisons, and discussion, about 15 mutations out of 30 are very useful in solving different types of optimization problems due to its correct and slow convergence with all dimensions. Meanwhile, the others are not recommended to be used alone in designing DE algorithm. In fact, most of these mutations suffer from false convergence which may be classified as premature convergence and/or stagnation or even mixed convergence behavior in addition to the deterioration of performance as dimension increases. Thus, analysis of population diversity, success rate and convergence speed of mutation is a must to suggest an appropriate modification or proposing a novel idea for repairing by identifying the cause of defective or inefficiency. However, without these mutations, the innovation process might have been stopped and the top and successful mutations might not have been proposed. Finally, future research studies must focus on applying and experimentally investigate the proposed recommendations, insights, and guidelines to continue improving this research field. Furthermore, Future research studies may focus on applying these algorithms to solve constrained, multi-objective and large-scale benchmark optimization problems. **It is highly noted that the empirical analysis of this study may differ on another benchmark set according to no-free-lunch theorem.**

## REFERENCES

- [1] R. Storn and K. Price, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1995.
- [2] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
- [3] Y. Wang, H.-X. Li, T. Huang, and L. Li, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Appl. Soft Comput.*, vol. 18, pp. 232–247, May 2014.
- [4] X. Li and M. Yin, "Modified differential evolution with self-adaptive parameters method," *J. Combinat. Optim.*, vol. 31, no. 2, pp. 546–576, Feb. 2016.
- [5] A. A. Hadi, A. W. Mohamed, and K. M. Jambi, "Single-objective real-parameter optimization: Enhanced LSHADE-SPACMA algorithm," in *Studies in Computational Intelligence*, vol. 906. Cham, Switzerland: Springer, 2021, pp. 103–121.
- [6] A. K. Mohamedak, A. W. Mohamed, E. Z. Elfeky, and M. Saleh, "Solving constrained non-linear integer and mixed-integer global optimization problems using enhanced directed differential evolution algorithm," in *Studies in Computational Intelligence*, vol. 801. Cham, Switzerland: Springer-Verlag, 2019, pp. 327–349.
- [7] A. W. Mohamed, A. A. Hadi, and K. M. Jambi, "Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization," *Swarm Evol. Comput.*, vol. 50, Nov. 2019, Art. no. 100455.
- [8] A. K. Mohamed, A. A. Hadi, and A. W. Mohamed, "Generalized adaptive differential evolution algorithm for solving CEC 2020 benchmark problems," in *Proc. 2nd Novel Intell. Lead. Emerg. Sci. Conf.*, 2020, pp. 391–396.
- [9] A. W. Mohamed, "Solving stochastic programming problems using new approach to differential evolution algorithm," *Egyptian Informat. J.*, vol. 18, no. 2, pp. 75–86, Jul. 2017.
- [10] A. W. Mohamed, A. K. Mohamed, E. Z. Elfeky, and M. Saleh, "Enhanced directed differential evolution algorithm for solving constrained engineering optimization problems," *Int. J. Appl. Metaheuristic Comput.*, vol. 10, no. 1, pp. 1–28, Jan. 2019, doi: [10.4018/IJAMC.2019010101](https://doi.org/10.4018/IJAMC.2019010101).
- [11] H. Zhu, Y. C. He, X. Z. Wang, and E. C. C. Tsang, "Discrete differential evolutions for the discounted 0-1 knapsack problem," *Int. J. Bio-Inspired Comput.*, vol. 10, no. 4, pp. 219–238, Jan. 2017.



- [12] A. W. Mohamed, "A new modified binary differential evolution algorithm and its applications," *Sohag J. Sci.*, vol. 1, no. 1, pp. 29–35, Sep. 2016.
- [13] N. Hachicha, B. Jarbouli, and P. Siarry, "A fuzzy logic control using a differential evolution algorithm aimed at modelling the financial market dynamics," *Inf. Sci.*, vol. 181, no. 1, pp. 79–91, Jan. 2011.
- [14] C.-R. Dong, W. W. Y. Ng, X.-Z. Wang, P. P. K. Chan, and D. S. Yeung, "An improved differential evolution and its application to determining feature weights in similarity-based clustering," *Neurocomputing*, vol. 146, pp. 95–103, Dec. 2014.
- [15] S. A. El-Quliti, A. H. M. Ragab, R. Abdelaal, A. W. Mohamed, A. S. Mashat, A. Y. Noaman, and A. H. Altalhi, "A nonlinear goal programming model for university admission capacity planning with modified differential evolution algorithm," *Math. Problems Eng.*, vol. 2015, pp. 1–13, May 2015.
- [16] S. A. El-Quliti and A. W. Mohamed, "A generalized national planning approach for admission capacity in higher education: A nonlinear integer goal programming model with a novel differential evolution algorithm," *Comput. Intell. Neurosci.*, vol. 2016, pp. 1–14, Jan. 2016.
- [17] S. A. H. ElQuliti and A. W. Mohamed, "A large-scale nonlinear mixed-binary goal programming model to assess candidate locations for solar energy stations: An improved real-binary differential evolution algorithm with a case study," *J. Comput. Theor. Nanosci.*, vol. 13, no. 11, pp. 7909–7921, Nov. 2016.
- [18] R. Tanabe and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, Jul. 2014, pp. 1658–1665.
- [19] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 107–125, Feb. 2008.
- [20] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009.
- [21] A. K. Mohamed and A. W. Mohamed, "Real-parameter unconstrained optimization based on enhanced AGDE algorithm," in *Studies in Computational Intelligence*, vol. 801. Cham, Switzerland: Springer-Verlag, 2019, pp. 431–450.
- [22] A. K. Mohamed, A. W. Mohamed, E. Z. Elfeky, and M. Saleh, "Enhancing AGDE algorithm using population size reduction for global numerical optimization," in *Proc. Adv. Intell. Syst. Comput.*, vol. 723, 2018, pp. 62–72.
- [23] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
- [24] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—An updated survey," *Swarm Evol. Comput.*, vol. 27, pp. 1–30, Apr. 2016.
- [25] Y. Song, D. Wu, A. Wagdy Mohamed, X. Zhou, B. Zhang, and W. Deng, "Enhanced success history adaptive DE for parameter optimization of photovoltaic models," *Complexity*, vol. 2021, pp. 1–22, Jan. 2021.
- [26] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: Algorithm jSO," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, San Sebastian, Spain, Jun. 2017, pp. 1311–1318.
- [27] L. Cheng, Y. Wang, C. Wang, A. W. Mohamed, and T. Xiao, "Adaptive differential evolution based on successful experience information," *IEEE Access*, vol. 8, pp. 164611–164636, 2020.
- [28] Z. Meng, Y. Chen, and X. Li, "Enhancing differential evolution with novel parameter control," *IEEE Access*, vol. 8, pp. 51145–51167, 2020.
- [29] P. N. Awad, M. Z. Ali, J. J. Liang, and B. Y. Qu, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization," Nanyang Technol. Univ., Singapore, Tech. Rep., 2016.
- [30] H. Y. Fan and J. Lampinen, "A trigonometric mutation operation to differential evolution," *J. Glob. Optim.*, vol. 27, no. 1, pp. 105–129, Sep. 2003.
- [31] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 1679–1696, Mar. 2011.
- [32] A. W. Mohamed and P. N. Suganthan, "Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation," *Soft Comput.*, vol. 22, no. 10, pp. 3215–3235, May 2018.
- [33] J. L. K. Price and R. Storn, *Differential Evolution—A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag, 2005.
- [34] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello, "A comparative study of differential evolution variants for global optimization," in *Proc. 8th Annu. Conf. Genetic Evol. Comput. (GECCO)*, vol. 1, 2006, pp. 485–492.
- [35] Z. Yang, X. Yao, and J. He, "Making a difference to differential evolution," in *Proc. Adv. Metaheuristics Hard Optim.* Berlin, Germany: Springer, 2007, pp. 397–414.
- [36] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Differential evolution with multiple strategies for solving CEC2011 real-world numerical optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2011, pp. 1041–1048.
- [37] C.-K. Ting and C.-H. Huang, "Varying number of difference vectors in differential evolution," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2009, pp. 1351–1358.
- [38] M. M. Ali, "Differential evolution with generalized differentials," *J. Comput. Appl. Math.*, vol. 235, no. 8, pp. 2205–2216, Feb. 2011.
- [39] B. Dorronsoro and P. Bouvry, "Improving classical and decentralized differential evolution with new mutation operator and population topologies," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 67–98, Feb. 2011.
- [40] S. Wang, Y. Li, and H. Yang, "Self-adaptive differential evolution algorithm with improved mutation mode," *Int. J. Speech Technol.*, vol. 47, no. 3, pp. 644–658, Oct. 2017.
- [41] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello, "Modified differential evolution for constrained optimization," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 25–32.
- [42] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.
- [43] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.
- [44] C.-W. Chiang, W.-P. Lee, and J.-S. Heh, "A 2-opt based differential evolution for global optimization," *Appl. Soft Comput.*, vol. 10, no. 4, pp. 1200–1207, Sep. 2010.
- [45] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 482–500, Apr. 2012.
- [46] Y. Zhou, X. Li, and L. Gao, "A differential evolution algorithm with intersect mutation operator," *Appl. Soft Comput.*, vol. 13, no. 1, pp. 390–401, Jan. 2013.
- [47] M. Z. Ali, N. H. Awad, and P. N. Suganthan, "Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization," *Appl. Soft Comput.*, vol. 33, pp. 304–327, Aug. 2015.
- [48] L. Cui, G. Li, Q. Lin, J. Chen, and N. Lu, "Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations," *Comput. Oper. Res.*, vol. 67, pp. 155–173, Mar. 2016.
- [49] A. W. Mohamed, H. Z. Sabry, and M. Khorshid, "An alternative differential evolution algorithm for global optimization," *J. Adv. Res.*, vol. 3, no. 2, pp. 149–165, Apr. 2012.
- [50] A. W. Mohamed, "An improved differential evolution algorithm with triangular mutation for global numerical optimization," *Comput. Ind. Eng.*, vol. 85, pp. 359–375, Jul. 2015.
- [51] L. Tang, Y. Dong, and J. Liu, "Differential evolution with an individual-dependent mechanism," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 560–574, Aug. 2015.
- [52] A. W. Mohamed and A. K. Mohamed, "Adaptive guided differential evolution algorithm with novel mutation for numerical optimization," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 2, pp. 253–277, Feb. 2019.
- [53] H. Wang, S. Rahnamayan, H. Sun, and M. G. H. Omran, "Gaussian bare-bones differential evolution," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 634–647, Apr. 2013.
- [54] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 99–119, Feb. 2011.
- [55] W. Gong and Z. Cai, "Differential evolution with ranking-based mutation operators," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2066–2081, Dec. 2013.
- [56] Y. Cai and J. Wang, "Differential evolution with neighborhood and direction information for numerical optimization," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2202–2215, Dec. 2013.
- [57] W.-J. Yu, J.-J. Li, J. Zhang, and M. Wan, "Differential evolution using mutation strategy with adaptive greediness degree control," in *Proc. Annu. Conf. Genetic Evol. Comput. (GECCO)*, Jul. 2014, pp. 73–79.
- [58] H. Sharifi-Noghabi, H. R. Mashhadi, and K. Shojaei, "A novel mutation operator based on the union of fitness and design spaces information for differential evolution," *Soft Comput.*, vol. 21, no. 22, pp. 6555–6562, Nov. 2017.



- [59] Y. Cai, Y. Chen, T. Wang, and H. Tian, "Improving differential evolution with a new selection method of parents for mutation," *Frontiers Comput. Sci.*, vol. 10, no. 2, pp. 246–269, Apr. 2016.
- [60] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, pp. 617–644, Dec. 2009.



**ALI WAGDY MOHAMED** received the B.Sc., M.Sc., and Ph.D. degrees from Cairo University, Egypt, in 2000, 2004, and 2010, respectively. He is currently an Associate Professor with the Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University. He is also an Associate Professor of statistics with the Wireless Intelligent Networks Center (WINC), Faculty of Engineering and Applied Sciences, Nile University. He has recognized among the top 2% scientists according to Stanford University report 2019. He has published more than 55 articles in reputed and high impact journals, such as Information Sciences, Swarm and Evolutionary Computation, Computers and Industrial Engineering, Intelligent Manufacturing, Soft Computing, and International Journal of Machine Learning and Cybernetics. He has supervised two Ph.D. and one master student. His research interests include mathematical and statistical modeling, stochastic and deterministic optimization, swarm intelligence and evolutionary computation, real world problems, such as industrial, transportation, manufacturing, education, and capital investment problems. He has been awarded the Publons Peer Review Awards 2018, for placing in the top 1% of reviewers worldwide in assorted field. He is also an Associate Editor of Swarm and Evolutionary Computation Journal (Elsevier). He is also an Editor of more than ten journals, such as Information Sciences, Applied Mathematics, Engineering, System Science, and Operations Research. He serves as a Reviewer for more than 70 international accredited top-tier journals. He has presented and participated in more than five international conferences. He participated as a member of the reviewer committee for 35 different conferences sponsored by Springer and IEEE. He has obtained Rank 3 in CEC'17 competitions on single objective bound constrained real-parameter numerical optimization, in Proceedings of IEEE Congress on Evolutionary Computation, IEEE-CEC 2017, San Sebastián, Spain. He has obtained Rank 3 and Rank 2 in CEC'18 competitions on single objective bound constrained real-parameter numerical optimization and competition on large scale global optimization, in Proceedings of IEEE Congress on Evolutionary Computation, IEEE-CEC 2017, Sao Paolo, Brazil. He has obtained Rank 2 in CEC'20 competitions on single objective bound constrained real-parameter numerical optimization, in Proceedings of IEEE Congress on Evolutionary Computation, IEEE-CEC 2020, Poland.



**ANAS A. HADI** received the M.Sc. and Ph.D. degrees in computer science from KAU, in 2013 and 2019, respectively. He is currently an Assistant Professor with the Computer Science Department, Faculty of Computing and Information Technology, King Abdulaziz University. His current research interests include machine learning, optimization, brain–computer interface (BCI), and feature selection.



**ALI KHATER MOHAMED** received the B.Sc., M.Sc., and Ph.D. degrees from the Department of Operations Research and Decision Support, Cairo University. He is currently an Assistant Professor with the Faculty of Computer Science, October University for Modern Sciences and Arts (MSA), Giza, Egypt. He is interested in statistics, mathematics, quantitative methods, operations research, and evolutionary algorithms. He serves as a reviewer for many reputable journals. He is the runner up in CEC2020 competition on single objective bound constrained real-parameter numerical optimization.

...