

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/309434512>

Fitness Proportionate Random Vector Selection based DE Algorithm (FPRVDE)

Article in International Journal of Advanced Computer Science and Applications · January 2016

DOI: 10.14569/IJACSA.2016.070946

CITATIONS

0

READS

19

3 authors:



Qamar Abbas

International Islamic University, Islamabad

14 PUBLICATIONS 40 CITATIONS

[SEE PROFILE](#)



Ahmad Jamil

Kohat University of Science and Technology

68 PUBLICATIONS 417 CITATIONS

[SEE PROFILE](#)



Hajira Jabeen

University of Bonn

52 PUBLICATIONS 241 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Biomedical signal processing using artificial intelligence [View project](#)



High resolution and concentrated time-frequency analysis tools using evolutionary computing techniques [View project](#)

Fitness Proportionate Random Vector Selection based DE Algorithm (*FPRVDE*)

Qamar Abbas

Department of Computing and
Technology
Iqra University
Islamabad, 44000, Pakistan

Jamil Ahmad

Computer Science Department,
Abasyn University,
Islamabad, 44000, Pakistan

Hajira Jabeen

Department of Computing and
Technology
Iqra University
Islamabad, 44000, Pakistan

Abstract—Differential Evolution (DE) is a simple, powerful and easy to use global optimization algorithm. DE has been studied in detail by many researchers in the past years. In DE algorithm trial vector generation strategies have a significant influence on its performance. This research studies that whether performance of DE algorithm can be improved by incorporating selection advancement in effective trial vector generation strategies. A novel advancement in DE trial vector generation strategies is proposed in this research to speeds up the convergence speed of DE algorithm. The proposed fitness proportion based random vector selection DE (FPRVDE) is based on the proportion of individual fitness mechanism. FPRVDE reduces the role of poor performing individuals to enhance its performance capability of DE algorithm. To form a trial vector using FPRVDE, individual based on the proportion of their fitness are selected. FPRVDE mechanism is applied to most commonly used set of DE variants. A comprehensive set of multidimensional function optimization problems is used to access the performance of FPRVDE. Experimental result shows that proposed approach accelerates DE algorithm.

Keywords—Differential Evolution; Fitness Proportion; Trial vector generation; Mutation; Optimization

I. INTRODUCTION

Differential evolution (DE), proposed by Storn and Price [1] is a stochastic population based evolutionary algorithm. The advantage of DE over other evolutionary algorithms is that it is simple, easy to use, speedy and greater probability of finding the global optima for function optimization [2] [3]. DE has been successfully used in various real life fields like Electrical power systems [4], electromagnetism [5], control systems [6], Bioinformatics [7], chemical engineering [8], image processing [9], artificial neural networks [10], signal processing [11] etc.

DE is a population based algorithm where a population of potential solutions is randomly initialized within an n-dimensional search space. All potential solutions are equally likely to be selected as parent in DE algorithm. The candidate solutions evolve themselves by exploring the entire search space over time to locate the optima of the objective function [12]. New vector is generated by adding the weighted difference between two population vectors to a third vector at each generation of DE algorithm [13]. Many algorithms are used for numerical benchmark optimization but DE has shown better performance than Genetic Algorithm and Particle Swarm Optimization for such problems [14] [15]. There are

many parameters in DE algorithm like Population 'NP', mutation probability 'F' and Crossover 'CR'. The DE algorithm mutation variants are formed by the linear combination of existing population members. The trial vector and target vector forms the mutant vector in DE. Throughout this paper x_i denotes the target vector (or current vector), u_i

represents the trial vector and v_i as a mutant vector. In DE algorithm, different mutation schemes are used to create the trial vector by using any combination of current, best and random vectors. The behavior of DE algorithm is influenced by the selection of mutation strategy and crossover scheme along with their control parameters: mutation probability 'F' and Crossover rate 'CR' [16] [17]. DE mutation strategies can be formed by the combinations of current vector, random vector (s), better vector and best vector. In any mutation strategy the order, number and name of vector(s) are very important. Throughout the following analysis x_g^{rk} states the r_k^{th} random vector for g^{th} generation, v_g^i will refer to i^{th} component of donor vector at g^{th} generation, x_g^{best} states the best vector at g^{th} generation, x_g^i refer to current vector at g^{th} generation. The most commonly used DE variants given in table-1 have been used by many researchers in their research work [18] [19].

TABLE I. COMMONLY USED DE VARIANTS

S.No	Name	Equation
V ₁	DE/rand/1/bin	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3})$
V ₂	DE/best/1/bin	$v_g^i = x_g^{\text{best}} + F(x_g^{r1} - x_g^{r2})$
V ₃	DE/rand/2/bin	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$
V ₄	DE/best/2/bin	$v_g^i = x_g^{\text{best}} + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4})$
V ₅	DE/rand to best/1/bin	$v_g^i = x_g^i + F(x_g^{\text{best}} - x_g^i) + F(x_g^{r1} - x_g^{r2})$
V ₆	DE/rand/1/exp	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3})$
V ₇	DE/best/1/exp	$v_g^i = x_g^{\text{best}} + F(x_g^{r1} - x_g^{r2})$

V_8	DE/rand/2/ exp	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$
V_9	DE/best/2/ exp	$v_g^i = x_g^{best} + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4})$
V_{10}	DE/rand to best/1/ exp	$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2})$

Qin et al. [20] have introduced self adaptive version of DE algorithm in their research work. They have control parameter self adaption as well as strategy self adaption in their paper. The memory concept based on finite learning period is used for self adaption. A strategy pool consisting of four commonly used mutation strategies “DE/rand/1/bin”, “DE/rand/2/bin”, “DE/rand-to-best/2/bin” and “DE/current-to-rand/1” is created for strategy self adaption. Zaharie [21] has introduced a parameter adaption scheme in this research that focuses on the choice of parameters by maintaining the diversity in the population. He has used $\lambda \in [0,1]$ as a coefficient of convex combination between random selected population and the best population member. Variance at component level for each population individual is calculated to maintain diversity in the population and population evolution is controlled by the evolution of variance. Liu and Lampinen [22] has introduced very famous parameter adaption method in their research work. Their parameter adaption scheme is based on the fuzzy controller designed for adaption of F and CR control parameters of DE algorithm. They have used two fuzzy logic controllers (FLC) to implement to adaptive the values of F and CR control parameters based on fuzzy control actions. Brest et al. [23] have introduced the idea of self adaption to choose the suitable values of DE control parameters F and CR. They have evolved both control parameters at individual level so that better individuals may survive and propagate to next generation. The evolution of F and CR is based on two predefined probabilities $\tau_1=0.1$, $\tau_2=0.1$ respectively. Zhang and Sanderson [24] have proposed adaptive differential evolution with optional external archive (JADE) parameter adaption based algorithm in their research work. They have used new strategy “DE/current to pbest/1” that is based on the conventional “DE/current to best/1” strategy. They generate new population with the union of current population and external archive containing inferior fit population members. They have adopted control parameter F by using Cauchy distribution and CR by using the normal distribution. They have also used two variants of JADE with name rand-JADE and nonaJADE. In rand-JADE instead of DE/current-to-pbest, DE/rand/1 is implemented & variant nona-JADE does not adopt adaptive parameter control instead it set values of both parameters to 0.5. Mallipeddi et al. [25] have introduced an ensemble of control parameter and mutations, crossover strategies in DE algorithm. A set of predefined values of F and CR control parameter values is used to form parameter pool and binomial, exponential versions of “DE/current-to-rand/1/bin” & JADE mutation strategies are used to form a strategy pool in their research work. A trial vector in EPSDE is generated based on assigned mutation strategy and assigned control parameter value from their corresponding pools. Wang, Cai and Zhand [26] have introduced a composite trial vector generation scheme in DE algorithm. They have used three various combination of F and CR $[F=1.0, Cr=0.1]$, $[F=$

$1.0, Cr=0.9]$ and $[F=0.8, Cr=0.2]$ that forms a strategy pool in composite DE (CoDE). Strategy pool in CoDE is formed by using four commonly used mutation strategies “rand/1/bin”, “current-to-rand/1” and “rand/2/bin”. A combination of control parameter values pool and a strategy from the strategy pool is selected to generate a trial vector against each target vector in the current population. Gong et al. [27] have introduced enhanced DE in their research work that is based on the strategy adaption mechanism (SaM). Further they ensembled SaM with JADE to form a strategy pool that is based on the convention mutation strategies of DE algorithm. DE/rand-to-pbest with archive, “DE/current-to-pbest” without archive, “DE/current-to-pbest” with archive, “DE/rand-to-pbest” without archive are used to form a strategy pool in enhanced DE. Das et al. [28] have proposed a neighbourhood base mutation strategy in their research work. They have used family of variants of “DE/target-to-best/1/bin” conventional mutation strategy based on the neighbourhood mechanism. A local and global neighbourhood scheme brings equilibrium in the exploration and exploitation without any extra load of other parameters. To generate a donor vector α and β are used as scaling factors for local and global neighbor respectively. Their proposed version has significant performance when compared with other DE variants. MinhazulIslam et al. [29] have introduced a novel crossover and mutation strategies in their research work. They have used a new mutation strategy “DE/current-to-gr_best/1” that utilizes q% of best population members and is based on the conventional strategy “DE/current-to-best/1”. The new crossover scheme in their research work is pbest crossover that utilizes p-top-ranked individual components. The third important aspect in their research is statistical distribution based adaption in DE control parameters F and CR.

DE mutation strategies are generated by using combination of random and/or best vector(s). In random selection best and worst members have same probability of selection as a parent. The worst parent may lead to worst child. A novel variant of DE is proposed in this research that will be helpful in avoiding the selection of bad performing individuals. The proposed variant will select those individuals having at least some level of fitness that will be helpful in enhancing the convergence speed and searching capability of DE algorithm. It is important to mention that fitter parent does not mean best parent fitter means better than worst performing members. This proposed variant will prove to be a significant addition in DE algorithm. The residue of the paper consist of the DE algorithm that is discussed in section-II, various crossover schemes are given in section-III of the paper, section-IV contains the proposed DE variant, the detail of benchmark functions is given in section-V of this paper and section-VI contains the results and discussion.

II. DE ALGORITHM

DE is a population based algorithm that consists of NP members where each population member is a point in a D-dimensional continuous hyperspace. D-dimensional real-valued parameter vectors in DE algorithm are initialized randomly in search of optimal solution. Vector $x_{i,G}$ represents a population member at G^{th} generation where $i=1, 2, 3, \dots, \dots$,

NP. In DE algorithm various operators like mutation, selection and crossover are used to create offspring population.

Mutation: In mutation a donor vector $v_{i,G+1}$ is generated against each target vector $x_{i,G}$ for i^{th} population member at each g^{th} iteration of DE algorithm. According to conventional DE mutation strategy *DE/rand/1* generates mutant vector $v_{i,G+1}$ by using the following equation

$$v_{i,G+1} = x_{r1,G} + F(x_{r2,G} - x_{r3,G}) \dots\dots\dots(1)$$

Where $x_{r1,G}$, $x_{r2,G}$ and $x_{r3,G}$ are randomly chosen population members such that $i \neq r_1 \neq r_2 \neq r_3$ and F is a difference vector scaling factor.

Crossover: In DE crossover operation the dimensions of target vector $x_{i,G}$ and a donor vector $v_{i,G+1}$ are swapped that is controlled by crossover control parameter CR. Crossover is used to enhance diversity in the DE population members. Binomial and exponential are two main crossover schemes of DE algorithm [16] [30] [31] that differ from each other in the way of number of mutation vector components distribution. Binomial vector generates the trial vector $u_{i,G} = (u_{i,1,g}, u_{i,2,g}, \dots, u_{i,D,g})$ by using following equation

$$u_{i,G} = \begin{cases} v_{i,j,G} & \text{if } (\text{randj}(0,1) \leq CR \text{ or } j = j_{\text{rand}}) \\ x_{i,j} & \text{otherwise} \end{cases} \dots\dots\dots(2)$$

Where j_{rand} is a random selected dimension from $[1, D]$, $v_{i,j,G}$ is the mutant vector; and $\text{randj}(0,1)$ a random value between 0 and 1, Control parameter CR takes value from (0, 1]. Whereas Exponential crossover scheme generates the trial vector $u_{i,G}$ by using following equation

$$u_{i,G} = \begin{cases} v_{i,j,G} & \text{for } j = \langle l \rangle_D + \langle l+1 \rangle_D + \dots + \langle l+L-1 \rangle_D \\ x_{i,j} & \text{otherwise} \end{cases} \dots\dots\dots(3)$$

Where modulo function is represented as $\langle \rangle_D$ with modulus D; $i = 1, 2, 3, \dots, NP$ and $j = 1, 2, 3, \dots, D$. A random number L is generated from $[1, D]$ and to represent a starting index l is chosen randomly from $[1, D]$.

Selection: In selection operation it is decided that whether target vector $x_{i,G}$ target vector or trial vector $u_{i,G}$ survives into next generation. The selection is done by using greedy selection based on the fitness of trial vector and target vector. The equation of selection operation is DE is given as follows

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & \text{if } (f(u_{i,G+1}) < f(x_{i,G})) \\ x_{i,G} & \text{otherwise} \end{cases} \dots\dots\dots(4)$$

Where $f(u_{i,G+1})$ is the fitness value of trial vector $u_{i,G}$ and $f(x_{i,G})$ is the fitness value of target vector $x_{i,G}$

III. PROPOSED FITNESS PROPORTIONATE RANDOM VECTOR SELECTION BASED DE ALGORITHM (FPRVDE)

The Proposed DE variant is based on the fitness proportionate random vector selection method that is used to select members from the parent population to generate the offspring population. Trial vector in FPRVDE is generated by selecting random individuals using fitness proportionate selection criteria. Fitness proportionate selection mechanism allocates the region to each individual according to its fitness value. To apply this method fitness of all individuals in the current population is calculated. Then population members are sorted based on the calculated fitness by maintaining population index of each target vector. After sorting population members, cumulative proportion C of each population member i is calculated. Then we iterate sorted population cumulative proportion that is greater than or equal to a random value r generated in the interval (0,1) to select as a fitness proportionate random vector. Then these fitness proportionate random vectors (*FPRV'S*) are used to form a trial vector against each target vector in the population. The fitness proportionate criteria will ensure that selected population members have some fitness level and are not the bad individuals of the current population. Fitness proportionate selection method helps in ignoring the poor performing population individuals as poor performing candidates may lead towards poor offspring's. The exceptional fit and inferior fit candidates will have equal chance of selection in *FPRVDE* version that will incorporate plenty of diversity in the population. The poor performing population members can generate poor offspring that might increase the convergence speed of DE algorithms while better performing population members will generate healthy offspring that might be helpful in improving the convergence speed of DE algorithm. The proposed variant maintains randomness in selecting the individuals since randomness is the important property in EC algorithm. *FPRVDE* ignores the worst performing members having about no fitness level and for other population member including weak, average and best performing population members have equal chance of selection in forming trial vector. If generated number is close to zero then less fitter population member be selected and if the random value is close to 1 then one of the best performing individual will be selected that incorporates reasonable randomness in *FPRVDE* version.

1. Generate the initial population $P_G = \{X_{1,G}, \dots, X_{NP,G}\}$ for generation $G=0$ and randomly initialize each population member $X_{i,G} = \{x_{i,G}^1, \dots, x_{i,G}^D\}$ where $i = 1, \dots, NP$
2. FOR $i = 1$ to NP
Calculate fitness $f(X_{i,G})$ for each population member $X_{i,G}$
END FOR
3. WHILE the stopping criterion is not true

Step 3.1 Mutation Step

/ Start of FPRVDE vectors selection */*

Step 3.1.1 FPRVDE vectors selection

Sort calculated fitness in ascending order by maintaining the index of each target vector $X_{i,G}$
FOR $i = 1$ to NP
Calculate the cumulative fitness proportion C of each target vector $X_{i,G}$

END FOR

FOR $i = 1$ to number of FPRVDE vectors

- Generate a random number r in the interval $(0,1)$
- Select population member i by iterating through sorted population until we reach at C that is greater than or equal to r
- Return i^{th} member index to be used as one of FPRVDE vectors in mutation strategy

END FOR

/ End of FPRVDE vectors selection */*

FOR $i = 1$ to NP

For the i^{th} target vector $X_{i,G}$ generate a donor vector $V_{i,G} = \{v_{i,G}^1, \dots, v_{i,G}^D\}$ with the specified mutation strategy (Any Strategy from table-II) with FPRVDE vectors

END FOR

Step 3.2 Crossover Step

FOR $i = 1$ to NP

For the i^{th} target vector $X_{i,G}$ generate a trial vector $U_{i,G} = \{u_{i,G}^1, \dots, u_{i,G}^D\}$ with the specified crossover scheme (Equation-2 or Equation-3)

END FOR

Step 3.3 Selection Step

FOR $i=1$ to NP

Evaluate the trial vector $U_{i,G}$ against the target vector $X_{i,G}$ with fitness function f

IF $f(U_{i,G}) \leq f(X_{i,G})$, THEN

$X_{i,G+1} = U_{i,G}$, $f(X_{i,G}) = f(U_{i,G})$

IF $f(U_{i,G}) \leq f(X_{best,G})$, THEN

$X_{best,G+1} = U_{i,G}$,

$f(X_{best,G}) = f(U_{i,G})$

END IF

END IF

END FOR

Step # 3.4 increment generation number $G=G+1$

Step 4. END WHILE

Fig. 1. Pseudocode of FPRVDE Algorithm

TABLE II. FPRVDE VERSIONS OF COMMONLY USED DE VARIANTS

S.No	Name	Equation
FPRVDE ₁	FPRVDE /rand/1/bin	$v_g^i = x_g^{fprv1} + F(x_g^{fprv2} - x_g^{fprv3})$
FPRVDE ₂	FPRVDE /best/1/bin	$v_g^i = x_g^{best} + F(x_g^{fprv1} - x_g^{fprv2})$
FPRVDE ₃	FPRVDE /rand/2/bin	$v_g^i = x_g^{fprv1} + F(x_g^{fprv2} - x_g^{fprv3}) + F(x_g^{fprv4} - x_g^{fprv5})$
FPRVDE ₄	FPRVDE /best/2/bin	$v_g^i = x_g^{best} + F(x_g^{fprv1} - x_g^{fprv2}) + F(x_g^{fprv3} - x_g^{fprv4})$
FPRVDE ₅	FPRVDE /rand to best/1/bin	$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{fprv1} - x_g^{fprv2})$
FPRVDE ₆	FPRVDE /rand/1/exp	$v_g^i = x_g^{fprv1} + F(x_g^{fprv2} - x_g^{fprv3})$
FPRVDE ₇	FPRVDE /best/1/exp	$v_g^i = x_g^{best} + F(x_g^{fprv1} - x_g^{fprv2})$
FPRVDE ₈	FPRVDE /rand/2/exp	$v_g^i = x_g^{fprv1} + F(x_g^{fprv2} - x_g^{fprv3}) + F(x_g^{fprv4} - x_g^{fprv5})$
FPRVDE ₉	FPRVDE /best/2/exp	$v_g^i = x_g^{best} + F(x_g^{fprv1} - x_g^{fprv2}) + F(x_g^{fprv3} - x_g^{fprv4})$
FPRVDE ₁₀	FPRVDE /rand to best/1/exp	$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{fprv1} - x_g^{fprv2})$

The mutation strategies given in table-II are FPRVDE mutation strategy versions corresponding to each commonly used DE mutation strategy given in table-I. Mutation strategies given in table-II used FPRVDE method to select each random vector in the current population.

IV. TEST FUNCTIONS AND EXPERIMENTAL RESULTS

A comprehensive set of 37 N-dimensional benchmark functions is used to evaluate the performance of proposed FPRVDE and commonly used set of DE variants. These benchmark functions are commonly used multidimensional global optimization problems. The necessary detail of these functions is given in table-III (a, b). The convergence speed of DE algorithm and the proposed variation are measured by using one of the most commonly used performance metric that is number of function calls (NFC) [32]. The acceleration rate is also calculated. Acceleration rate (AR) that is based on NFC is used to compare convergence speed of algorithm. Here AR for DE algorithm and proposed FPRVDE variation is defined by the following formula

$$AR = \frac{NFC_{DE}}{NFC_{FPRVDE}} \dots\dots\dots(5)$$

$AR > 1$ means that FPRVDE is faster than DE, $AR < 1$ means that FPRVDE is slower than DE and $AR = 1$ will shows that DE and FPRVDE have same convergence speed. Further the average acceleration rate and is calculated using equation (6) for the suit of functions used in this research. Average acceleration rate

$$A.A.R = AR_{ave} = \frac{1}{n} \sum_{i=1}^n AR_i \dots\dots\dots(6)$$

Experimental results reported in this section are generated for number of functions calls over a test suit of N-dimensional functions. NFC is the most common performance metric used in evolutionary algorithms [19] [33]. The results are generated using Exponential and binomial crossover schemes over 30 independent trials. NFC experimental results are reported in tables (II-VII) with best values as boldfaces. Convergence graphs of some functions are contained in Figure-2 and Figure-3 for DE and FPRVDE. Commonly used DE variants given in table-I as DE₁-DE₁₀ are corresponding AFPRVDE variants are reported in table-II as FPRVDE₁- FPRVDE₁₀. Experimental results are generated by using control parameter N_p (Population Size) with value 30 and dimensions are taken as 10D, 20D and 30D.. Control parameter F and CR values used are F=0.5, CR=0.9 [16] [30] [34]. Number of function calls value is calculated by taking 30 trials for both DE and FPRVDE for maximum NFC $10^4 \times \text{DIM}$ [35]. To find out NFC, VTR value is set to 0.0001 and Max-NFC values are 100,000; 200,000 and 300,000 for 10D, 20D and 30D respectively for both DE and FPRVDE in all functions.

TABLE III. (a)TEST BENCHMARK FUNCTION

Function	Name of Function (type)	Equation	Search Space	Optima
f_1	Sphere model (Separable, Multimodal)	$f(x) = \sum_{i=0}^n x_i^2$	$-5.12 \leq x_i \leq 5.12$	0
f_2	Axis parallel hyperellipsoid (Separable, Unimodal)	$f(x) = \sum_{i=0}^n i \cdot x_i^2$	$-5.12 \leq x_i \leq 5.12$	0
f_3	Schwefel's problem 1.2 (Non-Separable, Unimodal)	$f(x) = \sum_{i=0}^n \left(\sum_{j=0}^n x_j \right)^2$	$-65 \leq x_j \leq 65$	0
f_4	Rosenbrock's valley (Non-Separable, Unimodal)	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (1 - x_i)^2]$	$-30 \leq x_i \leq 30$	0
f_5	Rastrigin's function (Separable, Multimodal)	$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	$-5.12 \leq x_i \leq 5.12$	0
f_6	Griewangk's function (Non-Separable, Multimodal)	$f(x) = \sum_{i=1}^n \left(\frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \right) + 1$	$-600 \leq x_i \leq 600$	0
f_7	Sum of different power (Non-Separable, Multimodal)	$f(x) = \sum_{i=1}^n x_i ^{(i+1)}$	$-1 \leq x_i \leq 1$	0
f_8	Ackley's path function (Non-Separable, Multimodal)	$f(x) = -20 \exp \left(-0.2 \sqrt{\frac{\sum_{i=1}^n x_i^2}{n}} \right) - \exp \left(\frac{\sum_{i=1}^n \cos(2\pi x_i)}{n} \right) + 20 + e$	$-32 \leq x_i \leq 32$	0
f_9	Levy function (Separable, Multimodal)	$0.1 \left[\sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 \times (1 + \sin^2(3\pi x_{i+1})) + (x_n - 1)(1 + \sin^2(2\pi x_n)) \right]$	$-10 \leq x_i \leq 10$	0
f_{10}	Zakharov function (Non-Separable, Multimodal)	$f(x) = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5ix_i \right)^2 + \left(\sum_{i=1}^n 0.5ix_i \right)^4$	$-5 \leq x_i \leq 10$	0
f_{11}	Schwefel's problem 2.22 (Non-Separable, Unimodal)	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$-10 \leq x_i \leq 10$	0

f_{12}	Step function (Separable, Unimodal)	$f(x) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$	$-100 \leq x_i \leq 100$	0
f_{13}	Quartic function, i.e., noise (Separable, Unimodal)	$f(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1)$	$-1.28 \leq x_i \leq 1.28$	0
f_{14}	De Jong's function 4 (no noise) (Separable, Unimodal)	$f(x) = \sum_{i=1}^n ix_i^4$	$-1.28 \leq x_i \leq 1.28$	0
f_{15}	Alpine function (Separable, Multimodal)	$f(x) = \sum_{i=1}^n x_i \sin(x_i) + 0.1x_i $	$-10 \leq x_i \leq 10$	0
f_{16}	Pathological function (Non-Separable, Multimodal)	$f(x) = \sum_{i=1}^{n-1} \left(0.5 + \frac{\sin^2 \sqrt{(100x_i^2 + x_{i+1}^2) - 0.5}}{1 + 0.001(x_i^2 - 2x_ix_{i+1} + x_{i+1}^2)} \right)$	$-100 \leq x_i \leq 100$	0
f_{17}	Inverted cosine wave function (Non-Separable, Multimodal)	$f(x) = -\sum_{i=1}^{n-1} \left(\exp \left(\frac{-(x_i^2 + x_{i+1}^2 + 0.5x_ix_{i+1})}{8} \right) \times \cos \left(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5x_ix_{i+1}} \right) \right)$	$-5 \leq x_i \leq 5$	$-n+1$

TABLE III. (b)TEST BENCHMARK FUNCTION

f_{18}	Exponential problem (Non-Separable, Multimodal)	$f(x) = -\exp \left(-0.5 \sum_{i=1}^n x_i^2 \right)$	$-1 \leq x_i \leq 1$	-1
f_{19}	Levy and Montalvo Problem (Separable, Multimodal)	$f(x) = \left(\frac{\pi}{n} \right) \left(10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin 2(\pi y_{i+1})] \right) + (y_n - 1)$ where $y_i = 1 + \frac{1}{4}(x_i + 1)$	$-10 \leq x_i \leq 10$	0
f_{20}	Neumaier 2 Problem (Separable, Unimodal)	$f(x) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n (x_i x_{i-1})$	$-n^2 \leq x_i \leq n^2$	0
f_{21}	Salomon Problem (Non-Separable, Multimodal)	$f(x) = 1 - \cos(2\pi \ x\) + 0.1 \ x\ $ where $\ x\ = \sqrt{\sum_{i=1}^n x_i^2}$	$-100 \leq x_i \leq 100$	0
f_{22}	Cosine Mixture (Separable, Multimodal)	$f(x) = -0.1 \sum_{i=1}^n \cos(5\pi x_i) + \sum_{i=1}^n x_i^2$	$-1 \leq x_i \leq 1$	$-0.1x(n)$
f_{23}	Cigar (Separable, Multimodal)	$f(x) = x_1^2 + 100000 \sum_{i=1}^n x_i^2$	$-10 \leq x_i \leq 10$	0
f_{24}	Function '15' (Separable, Multimodal)	$f(x) = \sum_{i=1}^{n-1} [0.2x_i^2 + 0.1x_i^2 \sin(2x_i)]$	$-10 \leq x_i \leq 10$	0
f_{25}	Dixon Price (Non-Separable, Unimodal)	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^n i.(2x_i^2 - x_{i-1})^2$	$-10 \leq x_i \leq 10$	0
f_{26}	Ellipse Function (Separable, Unimodal)	$f(x) = \sum_{i=1}^n (10^{6(\frac{i-1}{n-1})} . x_i^2)$	$-100 \leq x_i \leq 100$	0
f_{27}	Tablet Function (Separable, Unimodal)	$f(x) = 10^4 x_1^2 + \sum_{i=2}^n x_i^2$	$-100 \leq x_i \leq 100$	0
f_{28}	Schewel (Separable, Multimodal)	$f(x) = \sum_{i=1}^n ((x_1 - x_i^2)^2 + (x_i - 1)^2)$	$-32 \leq x_i \leq 32$	0
f_{29}	Deflected Corrugated Spring (Separable, Multimodal)	$f(x) = 0.1 \sum_{i=1}^n \left((x_i - \alpha)^2 - \cos \left(K \sqrt{\sum_{i=1}^n ((x_i - \alpha)^2)} \right) \right)$	$0 \leq x_i \leq 10$ $K=5$ $\alpha=5$ $x_i \in [0, 2\alpha]$	0
f_{30}	Mishra 1 global optimization problem (Non-Separable, Multimodal)	$f(x) = (1 + x_n)^{x_n}$ where $x_n = n - \sum_{i=1}^{n-1} x_i$	$0 \leq x_i \leq 1$	2
f_{31}	Mishra 2 global optimization problem (Non-Separable, Multimodal)	$f(x) = (1 + x_n)^{x_n}$ where $x_n = n - \sum_{i=1}^{n-1} \frac{(x_i + x_{i+1})}{2}$	$0 \leq x_i \leq 1$	2

f_{32}	MultiModal global optimization problem (Separable, Multimodal)	$f(x) = \left(\sum_{i=1}^n x_i \right) \left(\prod_{i=1}^n x_i \right)$	$-10 \leq x_i \leq 10$	0
f_{33}	Plateau global optimization problem (Separable, Multimodal)	$f(x) = 30 + \sum_{i=1}^n \lfloor x_i \rfloor$	$-5.12 \leq x_i \leq 5.12$	30
f_{34}	Quintic global optimization problem (Separable, Multimodal)	$f(x) = \sum_{i=1}^n x_i^5 - 3x_i^4 + 4x_i^3 + 2x_i^2 - 10x_i - 4 $	$-10 \leq x_i \leq 10$	-1
f_{35}	Stochastic global optimization problem (Separable, Multimodal)	$f(x) = \sum_{i=1}^n \varepsilon_i \left x_i - \frac{1}{i} \right $	$-5 \leq x_i \leq 5$	0
f_{36}	Stretched V global optimization problem (Non-Separable, Multimodal)	$f(x) = \sum_{i=1}^{n-1} t^{1/4} \left[\sin(50t^{0.1}) + 1 \right]^2$ where $t = x_{i+1}^2 + x_i^2$	$-10 \leq x_i \leq 10$	0
f_{37}	XinSheYang (Non-Separable, Multimodal)	$f(x) = \frac{\left(\sum_{i=1}^n x_i \right)}{e^{\sum_{i=1}^n \sin(x_i^2)}}$	$-2\pi \leq x_i \leq 2\pi$	0

TABLE IV. 10D-NUMBER OF FUNCTIONS CALLS (NFC) AND ACCELERATION RATE (AR) OF DE AND FPRVDE VARIANTS

Function	rand/1/bin			best/1/bin			rand-to-best/1/bin			rand/2/bin			best/2/bin		
	DE ₁	FPRVDE ₁	A.R	DE ₂	FPRVDE ₂	A.R	DE ₃	FPRVDE ₃	A.R	DE ₄	FPRVDE ₄	A.R	DE ₅	FPRVDE ₅	A.R
f_1	217.03	177.20	1.22	104.83	91.57	1.14	439.70	344.13	1.28	273.07	224.27	1.22	102.70	88.10	1.17
f_2	206.97	169.67	1.22	103.97	89.23	1.17	400.77	321.87	1.25	254.87	207.20	1.23	103.07	87.47	1.18
f_3	1284.50	896.50	1.43	454.63	344.73	1.32	2973.07	2083.50	1.43	1668.87	1172.60	1.42	422.20	297.93	1.42
f_4	3078.30	2895.93	1.06	1305.44	1188.86	1.10	6460.33	4813.97	1.34	3665.60	2736.50	1.34	2827.29	1272.43	2.22
f_5	1427.69	605.80	2.36	-	-	-	5974.03	2277.79	2.62	2326.00	-	-	19333.00	13036.60	1.48
f_6	1955.07	693.13	2.82	356.50	-	-	12929.19	4091.82	3.16	1988.00	-	-	2964.86	1111.50	2.67
f_7	113.00	89.30	1.27	57.43	47.00	1.22	228.33	174.20	1.31	144.93	118.23	1.23	55.93	48.47	1.15
f_8	1928.69	573.50	3.36	-	-	-	7251.46	2819.07	2.57	3313.00	889.50	3.72	23812.33	18253.50	1.30
f_9	460.03	250.80	1.83	227.80	131.97	1.73	908.70	632.73	1.44	600.93	469.13	1.28	224.23	156.23	1.44
f_{10}	925.10	663.47	1.39	363.03	258.18	1.41	2052.00	1449.60	1.42	1167.30	826.13	1.41	338.43	226.40	1.49
f_{11}	397.23	320.23	1.24	195.30	200.89	0.97	819.43	639.13	1.28	520.43	398.20	1.31	200.33	164.83	1.22
f_{12}	148.43	106.93	1.39	71.17	57.24	1.24	304.10	196.30	1.55	187.43	131.10	1.43	69.17	57.32	1.21
f_{13}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{14}	93.43	71.70	1.30	45.60	34.83	1.31	177.90	130.87	1.36	113.30	83.70	1.35	45.50	34.43	1.32
f_{15}	18182.10	5376.40	3.38	16026.50	-	-	30839.63	15202.65	2.03	23863.37	8666.27	2.75	19874.47	17748.30	1.12
f_{16}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{17}	3766.47	889.50	4.23	490.00	-	-	21796.43	6421.33	3.39	10837.17	1260.00	8.60	9887.10	14297.04	0.69
f_{18}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{19}	221.87	169.17	1.31	101.83	84.57	1.20	478.30	363.40	1.32	290.40	221.63	1.31	111.17	92.73	1.20
f_{20}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{21}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{22}	219.20	177.23	1.24	105.93	90.39	1.17	481.80	364.67	1.32	292.33	232.43	1.26	105.23	89.73	1.17
f_{23}	407.60	330.70	1.23	193.27	169.20	1.14	844.73	651.33	1.30	511.90	416.37	1.23	189.97	160.73	1.18
f_{24}	216.53	173.00	1.25	103.60	88.90	1.17	439.60	342.00	1.29	268.57	221.20	1.21	100.80	84.77	1.19
f_{25}	-	-	-	203.00	-	-	-	-	-	-	-	-	234.00	15373.25	-
f_{26}	306.97	248.80	1.23	145.20	126.60	1.15	633.60	488.97	1.30	383.97	312.57	1.23	143.57	122.17	1.18
f_{27}	320.83	258.69	1.24	154.23	131.69	1.17	654.10	507.77	1.29	406.87	316.04	1.29	154.27	126.63	1.22
f_{28}	383.97	301.90	1.27	169.70	137.45	1.23	829.17	617.83	1.34	492.37	374.27	1.32	164.53	141.30	1.16
f_{29}	1537.37	849.27	1.81	1031.77	482.54	2.14	2632.30	2113.17	1.25	2133.60	1394.90	1.53	950.33	725.07	1.31
f_{30}	194.40	192.14	1.01	85.70	82.86	1.03	282.70	267.43	1.06	143.53	131.00	1.10	114.27	109.27	1.05
f_{31}	211.20	208.17	1.01	91.33	88.63	1.03	319.07	298.23	1.07	156.93	144.27	1.09	121.73	120.33	1.01
f_{32}	41.47	19.53	2.12	26.47	12.47	2.12	88.40	21.80	4.06	64.20	11.63	5.52	28.43	12.47	2.28
f_{33}	5.40	5.27	1.03	3.20	2.77	1.16	6.40	5.60	1.14	3.77	3.23	1.16	4.03	3.47	1.16
f_{34}	773.57	548.87	1.41	262.67	204.83	1.28	3417.97	1687.70	2.03	1008.60	644.93	1.56	624.57	1336.86	0.47
f_{35}	2017.40	902.47	2.24	400.50	453.00	0.88	-	-	-	24284.62	2791.10	8.70	413.83	445.68	0.93
f_{36}	31.60	23.20	1.36	26.00	6.14	4.23	46.97	31.00	1.52	38.13	22.95	1.66	32.70	25.53	1.28
f_{37}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

TABLE V. 10D-NUMBER OF FUNCTIONS CALLS (NFC) AND ACCELERATION RATE (AR) OF DE AND FPRVDE VARIANTS

Function	rand/1/exp			best/1/exp			rand-to-best/1/exp			rand/2/exp			best/2/exp		
	DE ₆	FPRVDE ₆	A.R	DE ₇	FPRVDE ₇	A.R	DE ₈	FPRVDE ₈	A.R	DE ₉	FPRVDE ₉	A.R	DE ₁₀	FPRVDE ₁₀	A.R
<i>f</i> ₁	195.63	184.57	1.06	140.73	134.20	1.05	266.77	247.83	1.08	216.20	207.03	1.04	145.77	136.97	1.06
<i>f</i> ₂	199.53	186.67	1.07	144.00	131.00	1.10	272.13	257.50	1.06	226.00	209.77	1.08	152.47	137.90	1.11
<i>f</i> ₃	1450.60	1248.77	1.16	808.47	663.65	1.22	2261.67	1968.10	1.15	1657.70	1412.10	1.17	765.83	594.90	1.29
<i>f</i> ₄	4457.47	4462.95	1.00	2813.57	1804.00	1.56	12168.87	6357.00	1.91	6153.20	3775.80	1.63	4416.00	2824.20	1.56
<i>f</i> ₅	389.14	345.07	1.13	261.53	225.00	1.16	587.17	537.03	1.09	469.00	426.00	1.10	1689.00	1597.00	1.06
<i>f</i> ₆	806.63	611.93	1.32	457.50	-	-	1736.17	1367.93	1.27	1137.37	818.40	1.39	904.08	1052.44	0.86
<i>f</i> ₇	122.60	115.30	1.06	88.47	75.30	1.17	163.13	146.03	1.12	134.87	122.10	1.10	95.93	83.83	1.14
<i>f</i> ₈	516.76	448.08	1.15	343.58	-	-	766.59	722.63	1.06	609.00	554.43	1.10	2055.00	2150.50	0.96
<i>f</i> ₉	140.03	118.97	1.18	108.77	98.97	1.10	196.67	164.10	1.20	160.57	155.50	1.03	117.97	108.43	1.09
<i>f</i> ₁₀	2040.73	1741.89	1.17	1163.27	858.67	1.35	3158.57	2738.23	1.15	2276.43	1694.73	1.34	1170.00	1044.23	1.12
<i>f</i> ₁₁	362.70	337.37	1.08	264.17	293.80	0.90	499.93	463.20	1.08	411.03	376.46	1.09	279.17	258.57	1.08
<i>f</i> ₁₂	129.07	118.70	1.09	95.47	87.37	1.09	174.27	161.40	1.08	147.87	130.77	1.13	99.50	88.10	1.13
<i>f</i> ₁₃	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₁₄	82.17	70.30	1.17	61.20	52.37	1.17	108.60	93.60	1.16	90.10	75.10	1.20	61.73	51.03	1.21
<i>f</i> ₁₅	8878.67	4080.68	2.18	6114.27	1666.29	3.67	7031.57	7788.83	0.90	7241.00	5948.28	1.22	6573.27	8645.10	0.76
<i>f</i> ₁₆	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₁₇	1026.23	804.15	1.28	501.27	416.50	1.20	2546.67	1811.28	1.41	1576.90	1072.00	1.47	2044.37	1781.07	1.15
<i>f</i> ₁₈	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₁₉	181.13	166.03	1.09	129.17	118.93	1.09	248.47	233.57	1.06	208.60	193.67	1.08	148.33	126.54	1.17
<i>f</i> ₂₀	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₂₁	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₂₂	191.27	178.67	1.07	136.53	128.17	1.07	268.03	253.20	1.06	221.50	207.23	1.07	147.93	137.60	1.08
<i>f</i> ₂₃	360.00	338.60	1.06	265.10	241.83	1.10	496.73	465.80	1.07	412.33	383.87	1.07	271.40	249.63	1.09
<i>f</i> ₂₄	191.37	180.23	1.06	138.43	130.93	1.06	259.43	247.63	1.05	217.30	200.87	1.08	145.07	134.30	1.08
<i>f</i> ₂₅	4753.55	3579.67	-	443.83	295.00	-	13984.04	8131.43	-	4080.56	2929.33	-	659.00	1738.25	-
<i>f</i> ₂₆	274.80	255.53	1.08	197.47	181.57	1.09	368.93	351.57	1.05	304.13	292.03	1.04	205.93	187.43	1.10
<i>f</i> ₂₇	293.07	290.00	1.01	211.70	191.80	1.10	405.17	356.03	1.14	329.60	286.18	1.15	221.27	190.43	1.16
<i>f</i> ₂₈	437.73	396.00	1.11	272.67	235.76	1.16	646.23	566.73	1.14	495.80	418.31	1.19	281.37	244.57	1.15
<i>f</i> ₂₉	248.10	213.90	1.16	202.67	206.17	0.98	352.57	305.67	1.15	240.20	272.57	0.88	174.40	198.70	0.88
<i>f</i> ₃₀	237.83	241.70	0.98	160.40	156.43	1.03	302.57	293.17	1.03	221.43	213.53	1.04	214.33	209.23	1.02
<i>f</i> ₃₁	250.33	252.23	0.99	170.90	166.18	1.03	314.57	312.30	1.01	237.07	225.87	1.05	224.83	222.73	1.01
<i>f</i> ₃₂	39.00	26.90	1.45	32.60	18.80	1.73	49.33	28.63	1.72	44.00	21.03	2.09	33.80	22.33	1.51
<i>f</i> ₃₃	6.73	5.93	1.13	4.90	4.70	1.04	7.30	6.67	1.10	5.47	5.27	1.04	5.93	5.50	1.08
<i>f</i> ₃₄	595.07	491.72	1.21	354.93	305.33	1.16	1101.90	873.30	1.26	660.27	567.66	1.16	985.30	1649.53	0.60
<i>f</i> ₃₅	5082.63	2678.23	1.90	819.12	713.54	1.15	-	-	-	-	9189.07	-	889.03	798.93	1.11
<i>f</i> ₃₆	102.37	57.96	1.77	72.37	26.25	2.76	61.07	82.25	0.74	88.93	45.00	1.98	90.60	60.93	1.49
<i>f</i> ₃₇	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

TABLE VI. 20D-NUMBER OF FUNCTIONS CALLS (NFC) AND ACCELERATION RATE (AR) OF DE AND FPRVDE VARIANTS

Function	rand/1/bin			best/1/bin			rand-to-best/1/bin			rand/2/bin			best/2/bin		
	DE ₁	FPRVDE ₁	A.R	DE ₂	FPRVDE ₂	A.R	DE ₃	FPRVDE ₃	A.R	DE ₄	FPRVDE ₄	A.R	DE ₅	FPRVDE ₅	A.R
<i>f</i> ₁	621.73	377.93	1.65	217.83	183.33	1.19	2189.40	1095.57	2.00	1042.77	654.30	1.59	201.07	159.97	1.26
<i>f</i> ₂	632.63	407.80	1.55	235.27	199.77	1.18	2103.73	1133.80	1.86	1031.43	679.50	1.52	212.30	175.70	1.21
<i>f</i> ₃	19967.33	6367.70	3.14	3036.20	1869.57	1.62	105853.73	31443.03	3.37	37375.10	14276.20	2.62	2350.90	1338.07	1.76
<i>f</i> ₄	9581.83	7559.90	1.27	3168.92	2730.91	1.16	37431.93	17153.43	2.18	16060.03	9343.85	1.72	3243.67	3148.86	1.03
<i>f</i> ₅	20874.18	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₆	1323.57	632.20	2.09	330.46	270.11	1.22	11182.82	2834.08	3.95	2365.38	1143.36	2.07	519.19	536.38	0.97
<i>f</i> ₇	261.57	132.67	1.97	94.33	66.82	1.41	883.63	383.20	2.31	438.83	240.27	1.83	80.20	63.90	1.26
<i>f</i> ₈	26503.47	-	-	-	-	-	-	-	-	120792.00	-	-	-	-	-
<i>f</i> ₉	590.90	413.20	1.43	248.53	228.27	1.09	2223.93	1000.23	2.22	1070.60	614.67	1.74	312.53	187.63	1.67
<i>f</i> ₁₀	7023.60	3550.40	1.98	1922.50	1421.86	1.35	26725.13	12014.97	2.22	11749.37	6397.53	1.84	1611.03	1028.97	1.57
<i>f</i> ₁₁	1093.77	662.45	1.65	403.03	-	-	3791.53	1968.19	1.93	1916.67	1123.67	1.71	388.37	293.23	1.32
<i>f</i> ₁₂	416.40	217.83	1.91	151.25	106.33	1.42	1494.87	558.60	2.68	739.93	349.64	2.12	144.47	96.38	1.50
<i>f</i> ₁₃	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₁₄	342.57	192.37	1.78	118.47	96.43	1.23	1188.87	536.67	2.22	557.57	325.80	1.71	104.77	83.10	1.26
<i>f</i> ₁₅	31342.53	27903.97	1.12	20916.47	12814.90	1.63	35370.63	29618.83	1.19	48597.90	34810.60	1.40	30609.34	13474.37	2.27
<i>f</i> ₁₆	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₁₇	110766.40	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₁₈	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₁₉	629.50	356.67	1.76	208.14	164.64	1.26	2488.83	1107.17	2.25	1158.63	653.80	1.77	202.43	170.22	1.19
<i>f</i> ₂₀	70111.12	82374.67	-	46788.86	40745.32	-	66372.56	69646.32	-	84393.60	84237.05	-	42102.07	33848.90	-
<i>f</i> ₂₁	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₂₂	644.30	376.87	1.71	219.83	176.73	1.24	2518.40	1180.20	2.13	1171.70	690.83	1.70	211.59	167.62	1.26
<i>f</i> ₂₃	1114.40	692.57	1.61	393.40	329.90	1.19	3969.23	2029.77	1.96	1890.80	1188.63	1.59	357.00	293.37	1.22
<i>f</i> ₂₄	606.67	373.00	1.63	216.03	178.10	1.21	2117.20	1085.00	1.95	1028.23	643.60	1.60	194.13	158.27	1.23
<i>f</i> ₂₅	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₂₆	841.20	527.07	1.60	298.67	249.77	1.20	3000.63	1539.57	1.95	1428.30	900.00	1.59	270.07	220.90	1.22
<i>f</i> ₂₇	862.27	535.55	1.61	309.83	251.53	1.23	3020.43	1544.30	1.96	1458.93	893.54	1.63	280.20	226.37	1.24
<i>f</i> ₂₈	1094.80	726.00	1.51	364.67	314.93	1.16	4289.73	1937.90	2.21	1927.87	1084.20	1.78	340.40	343.23	0.99
<i>f</i> ₂₉	2406.07	2429.43	0.99	1968.70	1011.03	1.95	8503.43	4641.87	1.83	5674.47	2957.97	1.92	1643.43	1320.17	1.24
<i>f</i> ₃₀	403.57	386.44	1.04	159.87	155.08	1.03	769.40	663.36	1.16	332.50	284.81	1.17	211.13	208.57	1.01
<i>f</i> ₃₁	426.53	404.00	1.06	166.97	161.53	1.03	839.37	702.86	1.19	347.87	300.78	1.16	220.23	216.17	1.02
<i>f</i> ₃₂	84.40	24.03	3.51	40.47	8.39	4.82	209.33	24.90	8.41	139.43	9.93	14.04	53.07	11.90	4.46
<i>f</i> ₃₃	2.17	2.00	1.08	1.90	1.70	1.12	2.10	2.20	0.95	2.43	1.53	1.59	1.77	1.80	0.98
<i>f</i> ₃₄	3095.53	1101.33	2.81	538.87	403.82	1.33	42457.20	5154.20	8.24	5264.00	1862.53	2.83	893.77	539.40	1.66
<i>f</i> ₃₅	-	4099.13	-	921.00	-	-	-	-	-	-	-	-	1618.00	-	-
<i>f</i> ₃₆	20.10	27.40	0.73	33.37	11.07	3.02	34.30	27.41	1.25	30.90	18.85	1.64	31.37	32.27	0.97
<i>f</i> ₃₇	108.13	20.43	5.29	74.80	8.50	8.80	99.37	50.00	1.99	146.70	12.60	11.64	93.50	62.50	1.50

TABLE VII. 20D-NUMBER OF FUNCTIONS CALLS (NFC) AND ACCELERATION RATE (AR) OF DE AND FPRVDE VARIANTS

Function	rand/1/exp			best/1/exp			rand-to-best/1/exp			rand/2/exp			best/2/exp		
	DE ₆	FPRVDE ₆	A.R	DE ₇	FPRVDE ₇	A.R	DE ₈	FPRVDE ₈	A.R	DE ₉	FPRVDE ₉	A.R	DE ₁₀	FPRVDE ₁₀	A.R
<i>f</i> ₁	422.57	397.20	1.06	331.80	318.97	1.04	576.47	546.63	1.05	498.23	477.47	1.04	332.07	318.37	1.04
<i>f</i> ₂	466.00	432.90	1.08	364.50	345.47	1.06	633.27	596.87	1.06	547.50	524.37	1.04	369.37	347.60	1.06
<i>f</i> ₃	6577.87	5523.23	1.19	3841.90	3335.84	1.15	10667.17	9143.73	1.17	7775.13	6758.10	1.15	3347.33	2803.83	1.19
<i>f</i> ₄	22255.50	20637.14	1.08	18310.23	16057.75	1.14	36948.63	31304.63	1.18	54445.10	52531.39	1.04	19624.30	9034.67	2.17
<i>f</i> ₅	808.17	717.82	1.13	629.45	567.00	1.11	1238.53	1131.90	1.09	1046.12	986.28	1.06	-	-	-
<i>f</i> ₆	1048.63	863.56	1.21	598.09	540.42	1.11	1987.50	1543.14	1.29	1344.82	954.78	1.41	807.17	712.77	1.13
<i>f</i> ₇	237.93	226.77	1.05	170.77	141.72	1.20	316.47	290.40	1.09	260.57	238.67	1.09	184.03	164.13	1.12
<i>f</i> ₈	1052.27	906.88	1.16	798.50	-	-	1573.57	1479.97	1.06	1333.45	1221.15	1.09	-	-	-
<i>f</i> ₉	232.47	214.07	1.09	191.37	182.07	1.05	319.90	298.07	1.07	278.43	259.77	1.07	192.20	184.43	1.04
<i>f</i> ₁₀	53471.30	72995.09	0.73	48702.13	-	-	66895.20	70378.48	0.95	59634.30	54576.00	1.09	47413.70	64533.50	0.73
<i>f</i> ₁₁	763.83	708.14	1.08	607.00	567.00	1.07	1057.77	976.50	1.08	914.37	844.47	1.08	621.53	585.00	1.06
<i>f</i> ₁₂	279.33	256.10	1.09	223.57	210.77	1.06	377.53	343.90	1.10	334.47	311.57	1.07	225.00	210.50	1.07
<i>f</i> ₁₃	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₁₄	204.50	178.23	1.15	162.13	150.63	1.08	273.47	242.47	1.13	242.90	212.53	1.14	160.40	140.33	1.14
<i>f</i> ₁₅	5473.67	4826.07	1.13	7207.83	8564.00	0.84	5199.23	8918.20	0.58	4477.77	5936.53	0.75	7005.37	5273.17	1.33
<i>f</i> ₁₆	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₁₇	2373.00	-	-	1286.42	-	-	5219.20	-	-	3647.53	-	-	7633.33	-	-
<i>f</i> ₁₈	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₁₉	365.27	335.53	1.09	291.53	270.13	1.08	504.70	469.83	1.07	434.03	408.70	1.06	314.13	273.89	1.15
<i>f</i> ₂₀	5309.73	6038.67	-	4429.33	8324.33	-	7208.03	7188.70	-	7141.77	6977.07	-	6038.87	3666.87	-
<i>f</i> ₂₁	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₂₂	412.00	386.07	1.07	326.30	312.60	1.04	571.30	543.03	1.05	499.27	472.70	1.06	338.67	318.30	1.06
<i>f</i> ₂₃	747.50	711.13	1.05	593.77	567.20	1.05	1036.50	977.50	1.06	891.60	847.07	1.05	596.17	566.10	1.05
<i>f</i> ₂₄	412.27	389.80	1.06	326.70	312.97	1.04	564.40	539.97	1.05	491.03	465.33	1.06	327.40	307.23	1.07
<i>f</i> ₂₅	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
<i>f</i> ₂₆	575.77	541.83	1.06	450.43	434.10	1.04	791.07	746.73	1.06	678.53	651.40	1.04	455.30	433.17	1.05
<i>f</i> ₂₇	593.97	588.81	1.01	471.80	434.90	1.08	820.90	738.20	1.11	709.23	648.07	1.09	477.10	429.67	1.11
<i>f</i> ₂₈	1289.80	1243.29	1.04	727.70	626.69	1.16	2204.03	2060.70	1.07	1595.73	1295.64	1.23	741.50	662.37	1.12
<i>f</i> ₂₉	291.67	240.73	1.21	208.37	215.87	0.97	396.60	312.10	1.27	355.60	285.80	1.24	229.53	187.27	1.23
<i>f</i> ₃₀	524.37	524.61	1.00	405.23	402.53	1.01	666.47	653.39	1.02	541.47	514.61	1.05	524.57	528.87	0.99
<i>f</i> ₃₁	536.57	542.73	0.99	415.37	408.18	1.02	680.97	661.90	1.03	555.87	531.04	1.05	540.00	542.50	1.00
<i>f</i> ₃₂	69.87	45.13	1.55	52.73	29.80	1.77	75.00	43.67	1.72	73.07	28.93	2.53	56.03	33.57	1.67
<i>f</i> ₃₃	3.23	2.13	1.52	2.13	1.93	1.10	1.93	2.03	0.95	2.23	2.47	0.91	2.63	2.07	1.27
<i>f</i> ₃₄	1262.93	1087.59	1.16	818.10	722.00	1.13	2230.93	1848.17	1.21	1493.03	1317.50	1.13	2706.10	2832.20	0.96
<i>f</i> ₃₅	-	837.33	-	3407.56	2860.25	1.19	-	-	-	-	-	-	4525.07	5111.53	0.89
<i>f</i> ₃₆	205.87	131.76	1.56	217.87	33.45	6.51	147.87	152.07	0.97	138.63	116.21	1.19	197.33	111.23	1.77
<i>f</i> ₃₇	11.23	8.20	1.37	10.57	3.27	3.23	14.13	7.57	1.87	10.83	4.27	2.54	10.87	8.30	1.31

TABLE VIII. 30D-NUMBER OF FUNCTIONS CALLS (NFC) AND ACCELERATION RATE (AR) OF DE AND FPRVDE VARIANTS

Function	rand/1/bin			best/1/bin			rand-to-best/1/bin			rand/2/bin			best/2/bin		
	DE ₁	FPRVDE ₁	A.R	DE ₂	FPRVDE ₂	A.R	DE ₃	FPRVDE ₃	A.R	DE ₄	FPRVDE ₄	A.R	DE ₅	FPRVDE ₅	A.R
f_1	1280.50	573.77	2.23	341.60	279.37	1.22	7559.47	2118.40	3.57	2631.80	1209.93	2.18	291.17	237.93	1.22
f_2	1361.63	640.70	2.13	378.13	309.50	1.22	7553.47	2294.87	3.29	2761.40	1326.23	2.08	325.77	264.37	1.23
f_3	181964.47	20298.43	8.96	9644.17	5503.69	1.75	-	140235.53	-	-	61517.93	-	6959.47	3422.93	2.03
f_4	21064.43	13081.17	1.61	5721.52	4751.62	1.20	154358.63	34695.17	4.45	42251.63	18119.76	2.33	4869.52	4828.53	1.01
f_5	152395.82	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_6	2089.50	828.86	2.52	481.21	391.60	1.23	19376.46	3412.00	5.68	4366.75	1782.19	2.45	520.71	345.57	1.51
f_7	519.67	160.53	3.24	124.47	90.50	1.38	2704.80	546.63	4.95	1005.13	364.30	2.76	103.00	72.30	1.42
f_8	196031.86	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_9	993.10	520.27	1.91	306.21	235.78	1.30	5285.90	1567.47	3.37	2164.73	1072.07	2.02	297.67	197.20	1.51
f_{10}	25485.97	8829.60	2.89	5023.00	3570.09	1.41	165938.20	37317.07	4.45	51840.63	19882.03	2.61	3920.97	2483.30	1.58
f_{11}	2217.33	981.00	2.26	615.20	379.07	1.62	12663.63	3693.00	3.43	4731.63	1944.93	2.43	568.30	402.70	1.41
f_{12}	873.60	317.77	2.75	235.33	-	-	5559.50	1012.47	5.49	1925.33	626.35	3.07	207.00	-	-
f_{13}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{14}	831.00	315.40	2.63	200.23	167.00	1.20	5180.97	1129.93	4.59	1699.23	684.27	2.48	171.50	147.33	1.16
f_{15}	38066.27	30367.10	1.25	18757.17	15014.47	1.25	45907.17	44991.87	1.02	56486.73	40908.67	1.38	30295.93	26086.03	1.16
f_{16}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{17}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{18}	770.03	146.73	-	33.17	26.90	-	3933.87	804.00	-	575.70	152.27	-	65.83	55.83	-
f_{19}	1334.93	520.17	2.57	321.34	255.84	1.26	9707.03	2109.23	4.60	3025.37	1221.10	2.48	293.00	263.24	1.11
f_{20}	113110.00	125993.48	-	100333.25	96550.33	-	121766.11	153223.33	-	104572.71	120941.94	-	105852.37	61268.83	-
f_{21}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{22}	1375.87	575.20	2.39	351.93	274.50	1.28	9378.57	2254.67	4.16	3062.33	1266.47	2.42	311.77	238.42	1.31
f_{23}	2312.27	1027.77	2.25	602.13	493.23	1.22	13693.33	3813.50	3.59	4765.67	2161.47	2.20	521.67	417.93	1.25
f_{24}	1252.67	559.70	2.24	332.97	273.33	1.22	7435.80	2092.00	3.55	2592.93	1199.77	2.16	289.33	234.97	1.23
f_{25}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{26}	1735.97	781.40	2.22	460.63	380.93	1.21	10468.83	2874.80	3.64	3585.30	1667.67	2.15	400.03	325.77	1.23
f_{27}	1767.03	789.43	2.24	475.27	384.70	1.24	10449.20	2889.17	3.62	3638.47	1675.37	2.17	410.67	327.63	1.25
f_{28}	2370.10	1207.93	1.96	606.70	564.70	1.07	15999.60	3732.33	4.29	4986.23	2046.57	2.44	580.53	577.83	1.00
f_{29}	3831.17	3826.07	1.00	2841.30	1220.33	2.33	26153.20	7957.40	3.29	11228.73	7069.67	1.59	2530.53	1245.53	2.03
f_{30}	639.27	583.45	1.10	226.73	-	-	1572.60	1173.90	1.34	574.27	-	-	297.10	-	-
f_{31}	666.53	607.88	1.10	230.70	-	-	1649.27	1246.65	1.32	601.47	8271.75	0.07	302.87	-	-
f_{32}	146.47	30.00	4.88	53.17	9.93	5.35	557.23	28.47	19.57	344.60	3.57	96.62	53.70	10.27	5.23
f_{33}	1.73	2.23	0.78	1.57	1.57	1.00	1.77	1.80	0.98	2.10	1.77	1.19	1.60	1.53	1.04
f_{34}	7678.30	1600.07	4.80	809.90	600.46	1.35	270693.67	8136.90	33.27	15289.67	3204.57	4.77	1079.87	604.57	1.79
f_{35}	-	10549.60	-	-	-	-	-	-	-	-	-	-	9549.81	-	-
f_{36}	34.63	21.26	1.63	18.57	9.00	2.06	28.40	19.38	1.47	26.07	30.50	0.85	40.53	30.50	1.33
f_{37}	11.43	13.03	0.88	12.40	12.83	0.97	12.67	12.27	1.03	14.40	12.20	1.18	13.60	16.60	0.82

TABLE IX. 30D-NUMBER OF FUNCTIONS CALLS (NFC) AND ACCELERATION RATE (AR) OF DE AND FPRVDE VARIANTS

Function	rand/1/exp			best/1/exp			rand-to-best/1/exp			rand/2/exp			best/2/exp		
	DE ₆	FPRVDE ₆	A.R	DE ₇	FPRVDE ₇	A.R	DE ₈	FPRVDE ₈	A.R	DE ₉	FPRVDE ₉	A.R	DE ₁₀	FPRVDE ₁₀	A.R
f_1	652.70	615.13	1.06	536.53	524.47	1.02	889.40	852.87	1.04	793.97	764.50	1.04	535.17	510.37	1.05
f_2	741.70	701.00	1.06	606.80	588.00	1.03	1015.70	970.93	1.05	892.90	865.57	1.03	600.33	575.13	1.04
f_3	14352.63	12284.50	1.17	8940.07	8221.74	1.09	23114.43	20119.90	1.15	17615.53	15490.23	1.14	7457.13	6480.43	1.15
f_4	42169.90	29721.00	1.42	52822.07	77288.78	0.68	71640.67	67045.45	1.07	123743.13	121478.96	1.02	81340.60	-	-
f_5	1248.69	-	-	1021.88	-	-	1881.83	-	-	1669.85	-	-	-	-	-
f_6	1188.27	1259.54	0.94	828.50	762.11	1.09	2602.27	1994.13	1.30	1691.17	1291.81	1.31	895.03	912.73	0.98
f_7	373.83	328.97	1.14	250.73	215.61	1.16	480.73	462.17	1.04	395.23	366.57	1.08	273.33	236.90	1.15
f_8	1590.63	1393.92	1.14	1305.67	-	-	2371.17	2220.00	1.07	2077.05	1955.88	1.06	-	-	-
f_9	343.00	327.03	1.05	289.07	282.03	1.02	469.43	441.50	1.06	420.43	389.93	1.08	286.70	282.80	1.01
f_{10}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{11}	1179.90	1092.79	1.08	972.90	868.76	1.12	1641.87	1517.75	1.08	1451.77	1347.72	1.08	992.50	925.97	1.07
f_{12}	434.67	400.60	1.09	363.10	341.10	1.06	588.00	539.17	1.09	538.10	487.53	1.10	360.93	340.17	1.06
f_{13}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{14}	336.03	299.97	1.12	280.37	261.50	1.07	460.10	403.20	1.14	403.10	368.90	1.09	271.40	245.60	1.11
f_{15}	3847.77	4484.40	0.86	6088.57	4451.20	1.37	5945.83	5311.87	1.12	4496.60	5532.27	0.81	4770.70	4713.03	1.01
f_{16}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{17}	3591.03	2610.68	1.38	2260.57	1512.00	1.50	8373.57	6694.38	1.25	5792.07	4362.94	1.33	26192.61	37701.14	0.69
f_{18}	104.00	100.10	-	88.03	79.87	-	112.67	109.43	-	104.50	103.07	-	126.70	127.73	-
f_{19}	548.77	511.40	1.07	452.00	433.77	1.04	762.90	716.20	1.07	672.50	642.83	1.05	472.07	438.79	1.08
f_{20}	10754.00	12386.73	-	11588.73	17553.27	-	12538.70	7911.57	-	12754.63	16314.73	-	7084.37	7028.27	-
f_{21}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{22}	637.87	603.60	1.06	527.13	511.50	1.03	893.37	852.40	1.05	796.03	758.27	1.05	536.90	513.23	1.05
f_{23}	1154.83	1089.77	1.06	947.80	915.33	1.04	1586.70	1509.67	1.05	1404.73	1346.47	1.04	941.67	904.53	1.04
f_{24}	643.83	609.87	1.06	531.10	512.57	1.04	876.60	836.90	1.05	778.30	744.43	1.05	525.00	502.00	1.05
f_{25}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
f_{26}	882.70	837.10	1.05	724.60	705.10	1.03	1214.77	1155.73	1.05	1079.67	1032.33	1.05	718.00	694.30	1.03
f_{27}	906.47	816.67	1.11	748.13	707.42	1.06	1244.70	1124.53	1.11	1099.50	1022.73	1.08	744.20	687.00	1.08
f_{28}	2318.87	2299.57	1.01	1400.00	1145.25	1.22	3705.17	5384.86	0.69	3282.33	7086.22	0.46	1353.97	1293.57	1.05
f_{29}	295.77	317.23	0.93	330.00	247.27	1.33	435.47	368.10	1.18	404.60	350.20	1.16	222.00	187.43	1.18
f_{30}	819.83	830.43	0.99	662.03	-	-	1038.23	1015.00	1.02	878.27	-	-	857.30	-	-
f_{31}	832.13	840.67	0.99	674.37	-	-	1056.53	1027.76	1.03	887.87	-	-	874.43	1617.12	0.54
f_{32}	85.77	870.33	0.10	73.33	18.40	3.99	100.23	256.57	0.39	94.97	17.70	5.37	74.40	43.70	1.70
f_{33}	2.20	1.93	1.14	1.70	1.67	1.02	1.70	1.83	0.93	1.60	1.33	1.20	1.30	1.50	0.87
f_{34}	1901.30	1674.28	1.14	1317.20	1209.78	1.09	3374.67	2895.40	1.17	2437.50	2149.50	1.13	4528.57	6533.87	0.69
f_{35}	16811.78	-	-	-	-	-	-	-	-	-	-	-	16593.92	9860.13	1.68
f_{36}	320.90	148.04	2.17	192.10	74.50	2.58	267.70	256.50	1.04	226.70	99.67	2.27	246.37	215.17	1.15
f_{37}	11.43	13.03	-	12.40	12.83	-	12.67	12.27	-	14.40	12.20	-	13.60	16.60	-

Experimental results given in tables (IV-IX) are obtained using the parameter settings given in section-IV of this paper. Results of number of function calls and acceleration rate performance parameters of each DE strategy and FPRVDE strategy are reported. For easy analysis, results of 10D, 20D and 30D are reported in separate tables. Tables IV-V contains 10-Dimensions results, tables VI-VII contains 20D results and tables VIII-IX contains 30D performance results of benchmark functions for each DE strategy in table-I and its corresponding FPRVDE strategy given in table-II. The best result of the NFC's for each function and each strategy pair (DE/FPRVDE) are highlighted in **boldface** and summary is reported in table-X. Few entries in the tables (IV, IX) are filled with dash symbol where variants fails to find any solution like none of the variant of DE or FPRVDE find any solution of function f_{16} and vice versa. For simple analysis the detailed summary of research results is presented in table-X. Table-IX contains the best values of NFC and average acceleration rate (A.A.R) for each DE and its corresponding FPRVDE mutation strategy. It is obvious from the research result that FPRVDE₁ has better convergence speed in most of the cases for 10D, 20D and 30D results. The average of average acceleration rate of FPRVDE mutation strategies than DE mutation strategies is **1.75**.

The performances of all FPRVDE (FPRVDE₁.... FPRVDE₁₀) mutation strategies dominate over all DE (DE₁.... DE₁₀) strategies for all variables (dimensions) for Separable functions $f_1, f_2, f_9, f_{19}, f_{22}, f_{23}, f_{24}, f_{26}, f_{27}$; non-Separable functions f_3, f_4, f_7, f_{14} ; unimodal functions; unimodal functions $f_2, f_3, f_4, f_{14}, f_{26}, f_{27}$ and multimodal functions $f_1, f_7, f_9, f_{19}, f_{22}, f_{23}, f_{24}$. For other separable/non-separable and unimodal/multimodal functions FPRVDE mutation strategies are better than DE strategies in most of the cases and DE strategies are better in few cases only.

None of the DE/FPRVDE variant reaches to VTR using the current parameter setting for separable & unimodal function f_{13} and for non-separable & multimodal functions f_{16}, f_{21} . For functions f_{20}, f_{37} , none of the algorithm reaches to VTR with few variables (10D) but converges to VTR with more variables (20D, 30D). A non-separable and multimodal function f_{18} converges to VTR only for more variables (30D).

Now we discuss the cases where the performance of DE mutation strategies is better than FPRVDE strategies. DE₁ is better than FPRVDE₁ for functions $f_5, f_8, f_{17}, f_{29}, f_{36}$ (for 10 variables) and functions f_5, f_8, f_{33}, f_{37} (for 20 variables); DE₂ is better than FPRVDE₂ for functions $f_{11}, f_{15}, f_{17}, f_{25}, f_{35}$ (for 10

variables), functions f_{11}, f_{33}, f_{35} (for 20 variables) and functions f_{30}, f_{31}, f_{37} (for 30 variables); DE₃ is better than FPRVDE₃ for a separable & multimodal function f_{33} (for 30 variables); DE₄ is better than FPRVDE₄ for function f_5 (for 10 variables), function f_8 (for 20 variables) and functions f_{30}, f_{31}, f_{36} (for 30 variables); DE₅ is better than FPRVDE₅ for functions $f_{17}, f_{34}, f_{35}, f_{36}$ (for 10 variables), functions $f_6, f_{28}, f_{33}, f_{35}, f_{36}$ (for 20 variables) and functions $f_{12}, f_{30}, f_{31}, f_{35}, f_{37}$ (for 30 variables); DE₆ is better than FPRVDE₆ for functions f_{30}, f_{31} (for 10 variables), functions f_{10}, f_{17}, f_{31} (for 20 variables) and functions $f_5, f_{15}, f_{29}, f_{30}, f_{31}, f_{32}, f_{35}$ (for 30 variables); DE₇ is better than FPRVDE₇ for functions f_6, f_8, f_{11}, f_{29} (for 10 variables), functions $f_8, f_{10}, f_{15}, f_{17}, f_{29}$ (for 20 variables) and functions $f_4, f_5, f_8, f_{30}, f_{31}$ (for 30 variables); DE₈ is better than FPRVDE₈ for functions f_{15}, f_{36} (for 10 variables), functions $f_{10}, f_{15}, f_{17}, f_{33}, f_{36}$ (for 20 variables) and functions $f_5, f_{28}, f_{32}, f_{33}$ (for 30 variables); DE₉ is better than FPRVDE₉ for functions f_{29} (for 10 variables), functions f_{15}, f_{17}, f_{33} (for 20 variables) and functions $f_5, f_{15}, f_{29}, f_{30}, f_{31}$ (for 30 variables); DE₁₀ is better than FPRVDE₁₀ for functions $f_6, f_8, f_{15}, f_{25}, f_{29}, f_{34}$ (for 10 variables), functions $f_{10}, f_{17}, f_{30}, f_{34}, f_{35}$ (for 20 variables) and functions $f_6, f_{17}, f_{30}, f_{31}, f_{33}, f_{34}, f_{37}$ (for 30 variables);

It can be summarized from number of functions calls parameter that in few cases FPRVDE is significantly worse and in most of the cases the performance of FPRVDE is better than DE algorithm variants for 10D, 20D and 30D respectively. Considering the acceleration rate parameter FPRVDE variants are faster as compared to DE variants in most of the cases while in few cases FPRVDE variants has significantly worst performance than DE variants.

Convergence graph are shown in Figure-2 and Figure -3 that contains average of best values of population members obtained at specific iterations during the evolutionary process. Convergence graph contains the fitness value against iteration in each sub-graph in Figure-2 and Figure-3 for 5000 iterations and 10 dimensions. From convergence graphs it can be observed that FPRVDE variants performs better than DE variants in most of the cases; in few cases FPRVDE has comparable performance while in some cases DE performance is better than FPRVDE. Convergence graph of few functions f_{15} (a₁-j₁), f_{17} (a₂-j₂), f_{35} (a₃-j₃), f_{37} (a₄-j₄) are given for 10 DE mutation strategies (DE₁....DE₁₀) and their corresponding FPRVDE (FPRVDE₁..... FPRVDE₁₀) mutation strategies. Convergence graphs of DE mutation strategies and its corresponding FPRVDE strategies are given in each sub-graph (a_i....j_i) in Figure-2 and Figure-3.

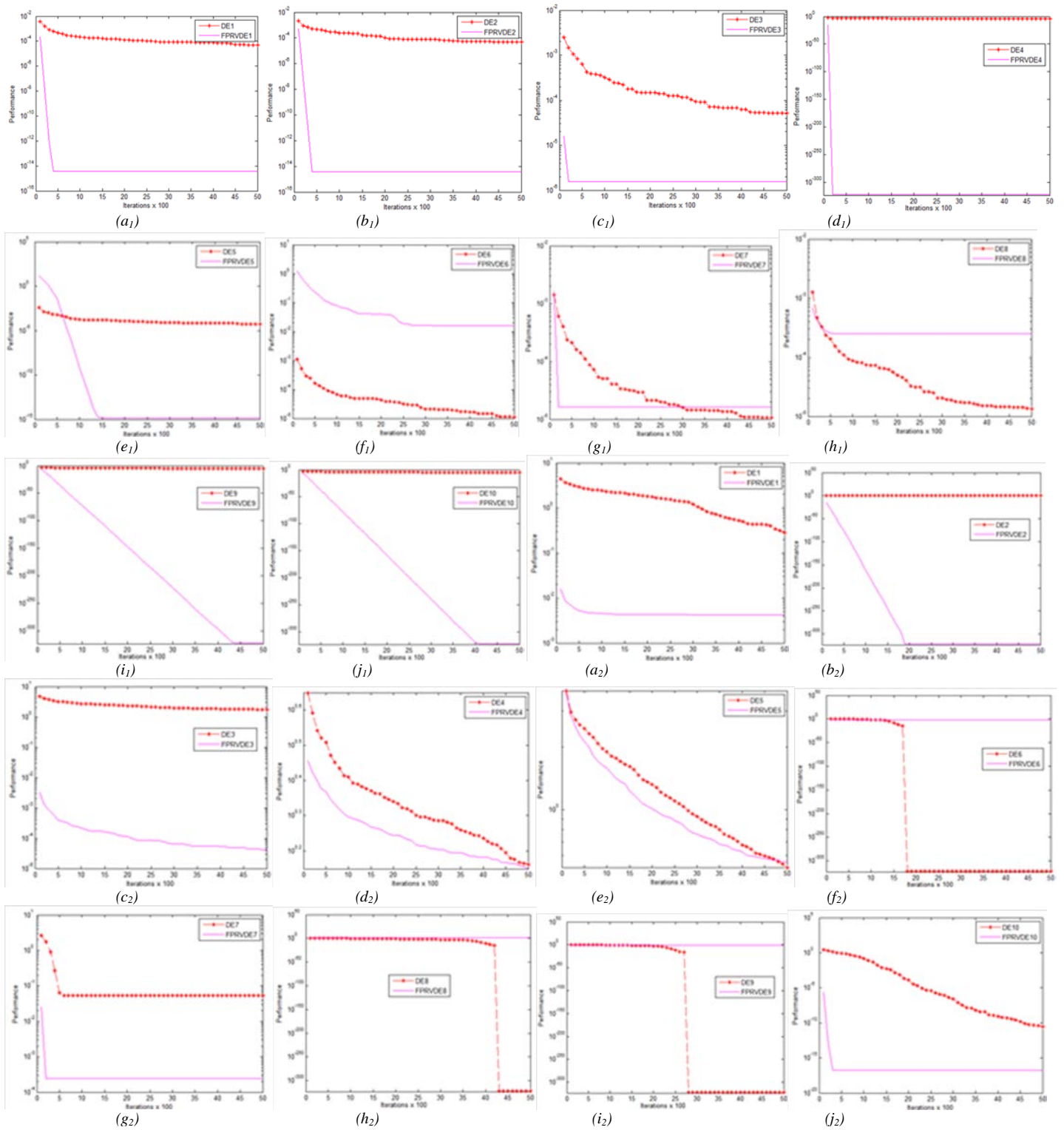


Fig. 2. DE and FPRVDE variants 10D Convergence graphs for function f_{15} (a1-j1), f_{17} (a2-j2)

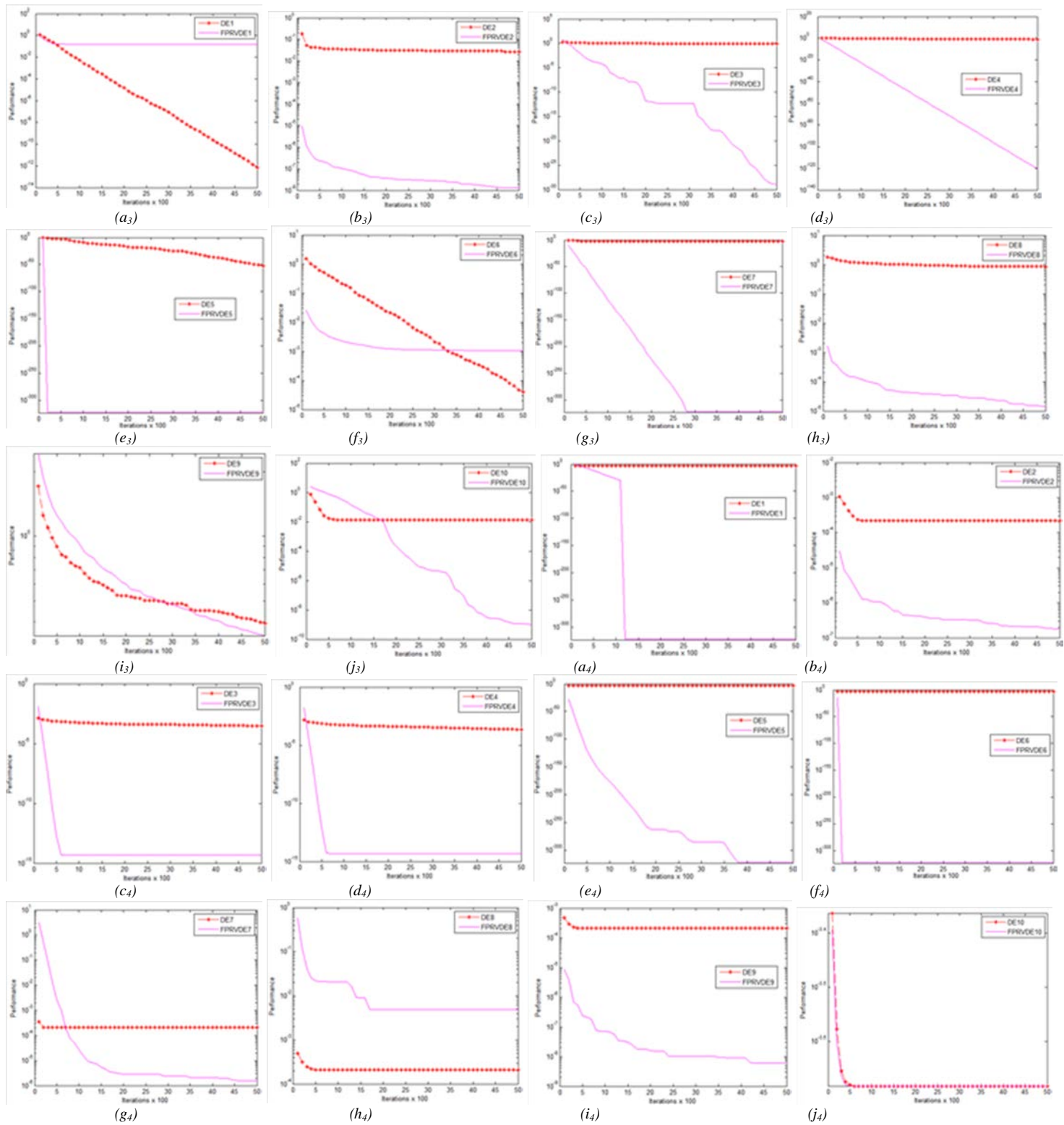


Fig. 3. DE and FPRVDE 10D Convergence graphs for function f_{35} (a₃-j₃), f_{37} (a₄-j₄)

TABLE XI. SUMMARY OF RESULTS NUMBER OF FUNCTIONS CALLS (NFC)
BEST AND AVERAGE ACCELERATION RATE (A.A.R)

10D			20D			30D		
Variant	Best	A.A.R	Variant	Best	A.A.R	Variant	Best	A.A.R
DE ₁	0	1.68	DE ₁	5	1.83	DE ₁	2	2.48
FPRVDE ₁	31		FPRVDE ₁	24		FPRVDE ₁	28	
DE ₂	6	1.39	DE ₂	2	1.77	DE ₂	3	1.61
FPRVDE ₂	23		FPRVDE ₂	25		FPRVDE ₂	23	
DE ₃	0	1.68	DE ₃	0	2.50	DE ₃	0	5.19
FPRVDE ₃	29		FPRVDE ₃	28		FPRVDE ₃	29	
DE ₄	2	2.12	DE ₄	1	2.57	DE ₄	3	5.98
FPRVDE ₄	28		FPRVDE ₄	27		FPRVDE ₄	25	
DE ₅	3	1.30	DE ₅	5	1.42	DE ₅	4	1.52
FPRVDE ₅	28		FPRVDE ₅	23		FPRVDE ₅	22	
DE ₆	2	1.21	DE ₆	3	1.13	DE ₆	8	1.08
FPRVDE ₆	29		FPRVDE ₆	28		FPRVDE ₆	22	
DE ₇	4	1.30	DE ₇	5	1.31	DE ₇	5	1.17
FPRVDE ₇	25		FPRVDE ₇	24		FPRVDE ₇	22	
DE ₈	2	1.15	DE ₈	5	1.10	DE ₈	4	1.05
FPRVDE ₈	28		FPRVDE ₈	25		FPRVDE ₈	26	
DE ₉	1	1.21	DE ₉	2	1.14	DE ₉	5	1.28
FPRVDE ₉	30		FPRVDE ₉	28		FPRVDE ₉	23	
DE ₁₀	5	1.10	DE ₁₀	5	1.16	DE ₁₀	7	1.05
FPRVDE ₁₀	26		FPRVDE ₁₀	24		FPRVDE ₁₀	21	

V. CONCLUSION AND FUTURE WORK

Trial vector has a key role in generating offspring/child population in DE algorithm. Different vectors like *random*, *best* and *current* are commonly used vectors to generate child population. Although random vector selection method is less biased and generates the more diverse population but has slow convergence to reach to a specific value VTR or optimal value. In this research a novel fitness proportionate based selection in random selecting random vectors used in DE (FPRVDE) mutation strategy is introduced. FPRVDE advancement is applied on most commonly used DE mutation strategies given in table-I. FPRVDE approach selects parent vectors by generating a random vector following the approach of fitness proportionate selection criteria. Performance of FPRVDE is accessed by taking a comprehensive set of multidimensional function optimization problems given in appendix section of this paper. Research result shows that FPRVDE variation approach enhances convergence speed of DE algorithm by maintaining appropriate altitude of diversity. The proposed approach ignores the poor performing individuals in generating the trial vector. NFC and AR performance parameters are used compare the performance of commonly used DE mutation strategies and FPRVDE approach. Research results shows that the convergence speed of FPRVDE approach is better than DE approach. NFC of FPRVDE mutation strategies is better than DE mutation strategies for most functions and various dimensions. Acceleration rate of FPRVDE mutation strategies for various functions and various mutation strategies is better than DE mutation strategies. In this research an effort is made to work in other directions of DE algorithm that will prove to a significant addition in DE research work. The future challenges of this research work can be the deep insight of FPRVDE along with its parameter settings and to explore the proposed approach in the other dimensions of research.

REFERENCES

- [1] R Storn and K Price, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," CA, Berkeley, Tech. Rep TR-95-012, 1995.
- [2] J. Brest and et. al., "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," IEEE Transaction on Evolutionary Computing, vol. 10, no. 6, pp. 646-657, December 2006.
- [3] K. Price, R. M. Storn, and J. A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series), 1st ed. New York:, USA: Springer-Verlag, 2005.
- [4] A. A. Abou El Ela, M. A. Abido, and S. R. Spea, "Optimal power flow using differential evolution algorithm," Electric Power Systems Research, vol. 80, no. 7, pp. 878-885, 2010.
- [5] K. A. Michalski, "Electromagnetic imaging of circular-cylindrical conductors and tunnels using a differential evolution algorithm," Microwave and Optical Technology Letters, vol. 27, no. 5, pp. 330-334, 2000.
- [6] R. Senkerik and et al., "Chaos driven Differential Evolution in the task of chaos control optimization," in IEEE Congress on Evolutionary Computation, Barcelona, Spain, 2010, pp. 1-8.
- [7] N. Noman and H. Iba, "Inference of gene regulatory networks using s-system and differential evolution," in In Proceedings of the 2005 conference on Genetic and evolutionary computation, 2005, pp. 439-446.
- [8] B. V. Babu, P. G. Chakole, and J. H. S. Mubeen, "Multiobjective differential evolution (MODE) for optimization of adiabatic styrene reactor," Chemical Engineering Science, vol. 60, no. 17, pp. 4822-4837, 2005.
- [9] M. Ali, C. W. Ahn, and M. Pant, "A robust image watermarking technique using SVD and differential evolution in DCT domain," Optik - International Journal for Light and Electron Optics, vol. 125, no. 1, pp. 428-434, January 2014.
- [10] N. Chauhan, V. Ravi, and D. K. Chandra, "Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks," Expert Systems with Applications, vol. 36, no. 4, pp. 7659-7665, May 2009.
- [11] B. Singh, B. Dhillon, and Y. S. Brar, "A Hybrid Differential Evolution Method for the Design of IIR Digital Filter.," International Journal on Signal & Image Processing, vol. 4, no. 1, pp. 1-10, January 2013.
- [12] X. Yao, Y. Liu, and G. Lin, "Evolutionary Programming Made Faster," IEEE Transaction on Evolutionary Computing, vol. 3, no. 2, pp. 82-102, July 1999.
- [13] S. F. P. Saramago G. T. T. Oliveira, "A Contribution to the Study About Differential Evolution ," Ciencia & Engenharia, vol. 16, no. 1/2, pp. 1-8, 2007.
- [14] S. Das, A. Abraham, and A. Konar, "Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives," Studies in Computational Intelligence, vol. 11, no. 6, pp. 1-38, 2008.
- [15] X. Xu and Y. Li, "Comparison between Particle Swarm Optimization, Differential Evolution and Multi-parents Crossover," in IEEE International Conference on Computational Intelligence and Security, 2007, pp. 124-127.
- [16] R Storn and Price K, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," Journal of Global Optimization, no. 11, pp. 341-359, 1997.
- [17] S. Das, Ajith Abraham, Uday K Chakraborty, and Amit Konar, "Differential Evolution Using a Neighborhood-Based Mutation Operator," IEEE Transactions on Evolutionary Computation, vol. 13, no. 3, pp. 526-553, June 2009.
- [18] A. Ghosh and et. al, "An improved differential evolution algorithm with fitness-based adaptation of the control parameters," Information Sciences, vol. 181, pp. 3749-3765, 2011.
- [19] S. M. Islam and et. al, "An Adaptive Differential Evolution Algorithm With Novel Mutation and Crossover Strategies for Global Numerical Optimization," IEEE transactions on systems, man, and cybernetics, vol. 42, no. 2, pp. 482-500, April 2012.

- [20] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 13, pp. 398-417, 2009.
- [21] D. Zaharie, "Control of population diversity and adaptation in differential evolution algorithms," in *In proceeding of MENDEL*, vol. 9, 2003, pp. 41-46.
- [22] J. Liu and J. Lampinen, "A Fuzzy Adaptive Differential Evolution Algorithm," *Soft Computing*, vol. 9, pp. 448-462, 2005.
- [23] J. Brest and et al., "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646-657, 2006.
- [24] J. Zhang and A. C. Sanderson, "JADE: Adaptive Differential Evolution With Optional External Archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945-958, 2009.
- [25] R. Mallipeddi and et al., "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679-1696, 2011.
- [26] Y. Wang, Z. Cai, and Q. Zhang, "Differential Evolution With Composite Trial Vector Generation Strategies and Control Parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55-66, 2011.
- [27] W. Gong and et al., "Enhanced differential evolution with adaptive strategies for numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics- Part B: Cybernetics*, vol. 41, no. 2, pp. 397-413, 2011.
- [28] S. Das and et al., "Differential Evolution Using a Neighborhood-Based Mutation Operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526-553, 2009.
- [29] S. Minhazul-Islam and et al., "An Adaptive Differential Evolution Algorithm With Novel Mutation and Crossover Strategies for Global Numerical Optimization," *IEEE Transactions on Systems, Man, and Cybernetics-PART B: Cybernetics*, vol. 42, no. 2, pp. 482-500, 2012.
- [30] E. Mezura-Montes, J V Reyes, and C A Coello Coello, "A Comparative Study of Differential Evolution Variants for Global Optimization," in *Genetic and Evolutionary Computation Conference(GECCO)*, Washington, USA, 2006, pp. 485-492.
- [31] M. Ali, M. Pant, and A. Abraham, "Simplex Differential Evolution," *Acta Polytechnica Hungarica*, vol. 6, no. 5, pp. 95-115, 2009.
- [32] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-Based Differential Evolution," *IEEE transactions on evolutionary computation*, vol. 12, no. 1, pp. 64-79, February 2008.
- [33] Y. Zhou, X. Li, and L. Gao, "A differential evolution algorithm with intersect mutation operator," *Applied Soft Computing*, vol. 13, pp. 390-401, 2013.
- [34] J. Brest, "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 10, pp. 646-657, December 2006.
- [35] Y. Wang, Z. Cai, and Q. Zhang, "Enhancing the search ability of differential evolution through orthogonal crossover," *Information Sciences*, vol. 185, pp. 153-177, 2012.