Continuous Optimization using Evolutionary Computing: Advancements in Differential Evolution Algorithm for function Optimization and Data Classification

By:

Qamar Abbas

Reg No: 051-07-15627

Supervised By:

Dr. Jamil Ahmad

Vice Chancellor, Abasyn University

Peshawar.

Co-Supervisor:

Dr. Hajira Jabeen

Assistant Professsor, Iqra University

Islamabad.

DEPARTMENT OF COMPUTING AND TECHNOLOGY

IQRA UNIVERSITY ISLAMABAD CAMPUS.

DECLARATION

I hereby declare that the work done in this research is my original piece of work performed under the supervision of Dr. Jamil Ahmad Vice Chancellor, Abasyn University Peshawar. Material from other authors, researchers has been properly acknowledged where ever included in this report. And further I also certify that no part of this work, separately or as a whole has been presented for the award of any other degree program.

....

PhD (CS)

APPROVAL SHEET

Continuous Optimization using Evolutionary Computing: Advancements in Differential Evolution Algorithm for function Optimization and Data Classification

By:

Qamar Abbas

Reg No: 051-07-15627

Accepted by the department of Computing and Technology, Iqra University, in partial fulfillment of the requirements for the award of the degree of

PhD in Computer Science

Supervisor: ____________________

Internal Examiner: ________________

External Examiner: _________________

Dean __________________

Department of Computing and Technology,
Iqra University, Islamabad Campus,
H-9, Islamabad.

ABSTRACT

Evolutionary computing algorithms have been implemented successfully for optimization problems. Differential Evolution (DE) is one of the evolutionary global optimization algorithm which has enjoyed considerable interest by many researchers in the recent years. A number of variants have been proposed to improve the performance of DE. However, most of the variants suffer from the problems of convergence speed and local optima. A novel tournament based parent selection mutation strategy of DE algorithm (TSDE) is proposed in this research. The proposed mutation strategy enhances searching capability in terms of fitness and improves convergence speed of the DE algorithm in terms of number of function calls. This research work also presents statistical comparison of existing DE mutation variants, which categorizes these variants in terms of their overall performance. The proposed mutation strategy is tested for standard benchmark functions and validated to train the artificial neural network for data classification problem. This thesis also introduces random controlled pool base differential evolution algorithm (RCPDE). A mutation strategy pool and a control parameter pool are used in RCPDE. The mutation strategy pool contains mutations strategies having diverse characteristics and control parameter pool contains varying nature pairs of control parameter values. The author has also observed that addition of rarely used control parameter values in the parameter pool and mutation strategy in the strategy pool is helpful to enhance the average fitness value and the number of function call performance parameters of DE algorithm. The proposed mutation strategies pool and control parameters pool in RCPDE are helpful in improving the solution quality and convergence speed of DE algorithm. RCPDE algorithm is tested over a test set of multi dimensional (N-dimensional) benchmark functions that shows significant performance of the proposed algorithm over many state of the art DE algorithms. To validate the performance of RCPDE algorithm; it is used to train artificial neural network for data classification problem.

Due to intensive study of DE algorithm by researchers; a number of mutation variants have been established for this algorithm. These mutation variants make DE algorithm more applicable, but due to the random development of these variants have created inconsistencies such as naming and formulation. Therefore, this research work also aims to identify inconsistencies and propose solution to make them consistent. Most of the inconsistencies exist because of the uncommon nomenclature used for these variants. In this research a

comprehensive study is carried out to identify inconsistencies in the nomenclature of mutation variants that does not match each other. Their proper and consistent names are proposed which provide significant contribution to the literature. The proposed names are assigned for conflicting variants that is based on the name of the variant, total number of vectors used to generate the trial vector and the order of the vectors to form the equation of these mutation variants. For effective conflict analysis of mutation strategies, trial vector generation mechanism of each variant is illustrated graphically. The consistent set of mutation variants will prove to be a valuable addition to DE literature.

DEDICATION

To hands,

Shivering and uplifted

Eyes heavy and thoughtful

Of my parents;

Hands ever praying for me.

Eyes with dreams in of my bright tomorrow

These hands may never fall down.

These eyes may never go to asleep.

This project is dedicated to my most respectful and honorable Parents and Teachers.

ACKNOWLEDGEMENTS

Up and above everything, I am grateful to almighty ALLAH, the beneficent, The Merciful, and His Prophet (Peace be upon him) who is forever a true torch of guidance for whole humanity. I am greatly obliged to “ALLAH” by whom grace I have been able to complete this project successfully.

This dissertation describes research undertaken at the department of Computing and Technology Iqra University, Islamabad Campus, between September 2010 and October 2015, under the supervision of Prof. Dr. Jamil Ahmad, to whom I am indebted for suggesting the subject and for his encouragement, guidance and support throughout the research program. His advice, discussion, and effective comments were always a source of motivation. I am also very thankful to my co-supervisor Dr. Hajira Jabeen for her and friendly discussion, keen interest and inspiring guidance.

I am grateful to my loving parents, my elder brother and my friends for providing me all sorts of moral and social support in life. Their prayers have enabled me to reach this stage.

I acknowledge the enabling role of the Higher Education Commission of Pakistan and appreciate the financial support through its Indigenous Ph.D. Scheme.

Table of Contents

List of Figures

List of Tables

List of Symbols, Abbreviations and Mathematical Notations

ʎ	–	Difference vector amplification factor
F	–	Mutation Probability
CR	–	Crossover Rate
NP	–	Population Size
DV	–	Difference Vector

List of Appendices

List of Publications

JOURNAL PAPERS:

[1]. **Q. Abbas,** J. Ahmad, H. Jabeen "A Novel Tournament Selection based Differential Evolution Variant for Continuous Optimization Problems", *Mathematical Problems in Engineering*, Vol. 5,No. 7, July 2015. [ISI-Indexed - I.F 0.76]

[2]. **Q. Abbas,** J. Ahmad, H. Jabeen, "The analysis, identification and measure to remove inconsistencies from Differential Evolution Mutation Variants", International conference on computer engineering and mathematical science (ICCEMS), pp.25–41, 2014.(Forwarded to Science Asia Journal for Publication (IF. 0.43))

[3]. W. H. Bangyal, J. Ahmad, I. Shafi, **Q. Abbas,** "A Forward Only Counter Propagation Network based Approach for Contraceptive Method Choice Classification Task", *Journal of Experimental & Theoretical Artificial Intelligence*, June 2012. [ISI-Indexed - I.F 0.67]

[4]. **Q. Abbas,** J. Ahmad, H. Jabeen, "Random Controlled Pool base Differential Evolution algorithm (RCPDE)", (Under Review in "Autosoft Journal" (IF. 0.218))

[5]. **Q. Abbas,** J. Ahmad, H. Jabeen, "Overture and Assessment of Adaptive and Laplace Mutation Operators in Differential Evolution (DE)", (Under Review in "Applied Computational Intelligence and Soft Computing" Journal (ISI Indexed))

[6]. **Q. Abbas,** J. Ahmad, H. Jabeen, "Fitness Proportionate Random Vector selection based DE algorithm (FPRVDE)", (Under Review in "Computational Intelligence and Neuroscience" (I.F. 0.596))

CONFERENCE PAPERS:

[1]. **Q. Abbas**, W. H. Bangyal, J. Ahmad, "Momentum term heals the performance of Back Propagation Algorithm for Digit Recognition", In Proceeding of International Conference on Emerging Technologies (ICET)/Pakistan,IEEE, pp.16–20,2010.

[2]. **Q. Abbas**, W. H. Bangyal, J. Ahmad, "Analysis of Learning Rate using BP Algorithm for Hand Written Digit Recognition Application", In Proceeding of International Conference on Information and Emerging Technologies (ICIET)/Pakistan, IEEE, pp. 1–5, 2010.

[3]. **Q. Abbas**, W. H. Bangyal, J. Ahmad, "Dynamic Hidden Layers Selection of ANN Architecture Using Particle Swarm Optimization", In Proceeding of International Conference on International Conference on Intelligence and Information Technology/Pakistan,IEEE, pp.418-420,2010.

[4]. A. Rauf,N. Kazim, **Q. Abbas**, S. Rehman, A. Ali, "Evolutionary based Automated Coverage Analysis for GUI Testing", In Proceeding of International Conference on Software Engineering and Data Mining (SEDM)/China,IEEE,pp. 219-224,2010.

[5]. W. H. Bangyal, J. Ahmad, **Q. Abbas**, "Forward Only Counter Propagation Network for Balance Scale Weight & Distance Classification Task", In Proceeding of International

Conference on Nature and Biologically Inspired Computing (NABIC)/Spain, IEEE, pp. 342-347,2011.

[6]. W. H. Bangyal, J. Ahmad, **Q. Abbas**, "Analysis of Learning Rate using CPN Algorithm for Hand Written Character Recognition Application",In Proceeding of International Conference on International Conference on Intelligence and Information Technology/Pakistan, IEEE, pp. 19-22,2010.

[7]. W. H. Bangyal, J. Ahmad, **Q. Abbas**, "Application of Counter Propagation Network in Handwritten Character Recognition Using the Winner-Take-All (WTA) Strategy", In Proceeding of International Conference on International Conference on Intelligence and Information Technology/Pakistan, IEEE, pp. 436-438, 2010.

[8]. M. Imran, H. Jabeen, M. Ahmad, **Q. Abbas**, W.H.Bangyal,"Opposition based PSO and mutation operators", In Proceeding of International Conference on Education Technology and Computer,(ICETC)/ China, IEEE, pp. 506-508, 2010.

[9]. M. Imran, Z. Manzoor, S. Ali, **Q. Abbas**, "Modified Paricle Swarm Optimization using Student t mutation", In Proceeding of International conference on computer networks and information technology,(ICCNIT)/Pakistan, IEEE, pp. 283 - 286, 2011.

[10]. A.Basit, J. Ahmad, I. Shafi, F. Riaz, **Q.Abbas**, W.H. Bangyal "Quality of service optimization of femtocells network using genetic algorithm", In Proceeding of International Multi topic Conference (INMIC)/Pakistan, IEEE, pp. 312 – 315, 2012.

Chapter 1: Introduction

1.0 Chapter Summary

This chapter presents an introduction to the dissertation. Major contribution, problem statement, brief background and organization of the thesis are also presented in this chapter.

1.1 Introduction and Background and

Function optimization is a process of finding optimal value by exploring the given feasible region. A large group of researchers is carrying out research on unimodal and multimodal optimization problems. As the conduct of research and new problem areas, the global optimization problems under investigation are becoming more complex. Fields of application of global optimization include Electrical power systems **[**1**]**, electromagnetism [2], control systems [3], Bioinformatics **[**4**]**, chemical engineering **[**5**]**, image processing [6], artificial neural networks [7], signal processing [8] etc.

1.2 Problem Statement

This dissertation presents some advancement in Differential Evolution (DE) algorithm to solve continuous optimization problems. The performance enrichment of DE algorithm is achieved by incorporating new parent selection schemes to generate trial vector. The proposed enhancement is capable of solving various optimization problems like non linear function optimization and artificial neural network training. For metaheuristic algorithms such optimization problems are still challenging to find optimal solution over time. Finding the true optimum value for the evolutionary computing algorithms is the challenging job. Differential Evolution algorithm can get stuck in the local optimum due to lack of exploration, diversity issue and local search issue in the infinite search space [9-13]. Detail of these issues is given as follows.

- <u>Slow convergence, diversity issue and local optima issue with DE mutation strategies:</u>

  DE algorithm has number of mutation strategies. These mutation strategies have different performance for varying nature of problems like separable/non-separable, uni-modal/multimodal etc. DE mutation strategies use combination of various parent vectors in generating the mutant vector. Most of DE mutation strategies use conventional parent vectors like current vector, random vector(s), best vector or the better vector. In any mutation strategy of DE algorithm; number of parents, order of parents, vector to be perturbed, and number of difference vectors are important. Selection of conventional parents cause DE algorithm to get stuck in local optima or created diversity issue in current population that reduces the convergence speed of DE algorithm. Improving the convergence speed and solution quality is another challenging task of DE algorithm. The DE algorithm suffers in local optima will not be able to find optimal value that reduces the convergence speed.

- <u>Pool base slow convergence and local optima issue</u>
  DE algorithm is sensitive to its control parameters values and mutation strategies for diverse nature of problems. This varying nature of mutation strategies and control parameter values in DE algorithm results slow convergence; stuck DE algorithm in local optima and causes diversity issue in the current population. The effective and efficient combination of pool of mutation strategies and pool of control parameter values is a challenging job in DE algorithm.

- <u>Inconsistencies in DE mutation strategies:</u>
  Moreover, there are some irregularities in the literature of the DE algorithm mutation strategies with respect to naming and formulation of these strategies. This dissertation identifies mutation irregularities and proposes corrections to these irregularities.

1.3 Motivation

The following problems are main motivations of this research work.

(a) Exploring the entire search space has significant role in finding the global optima. Exploration ability issue may cause to stuck in local optima problem. So the exploration ability should be improved by incorporating some diversity in the population.

(b) Sometimes the actual optima may lie in the neighbourhood of the global optima that can lead to neighbourhood search issue which may improve the search capability of the EC algorithms. The exploitation capability of the algorithm should be improved to avoid neighbourhood search issue.

(c) Artificial Neural network training using gradient based algorithms is slow over time. Evolutionary algorithm have been successfully applied to train Artificial Neural net architecture however they further need to be enhanced to get better generalization ability.

(d) Numerous researchers have advanced the state of the art to improve the working and performance of DE. However, there are some conventional notations being implicitly used in the DE literature. These notations became irregular with respect to the naming and formulations of DE mutation variants. The naming and formulation of DE mutation variants should be consistent for the prosperity of DE algorithm.

(e) Brute force or conventional algorithms cannot search in infinite search space; it is possible by EC Algorithms.

1.4 Major Contributions

<u>The main contribution of this dissertation is as follows:</u>

(1) Improved the convergence speed and solution quality over the feasible region by proposing a trial vector generation mutation strategies that employs a novel parent selection method.

(2) A novel pool based enhancement in DE algorithm is introduced that is helpful to incorporated diversity and enhance convergence speed of DE algorithm by utilizing efficient and effective pool of mutation strategies and control parameter.

(3) Tackled the problem of training artificial neural network and getting optimal values of weight vectors to improve its generalization ability.

(4) The irregularities in the literature of the DE algorithm mutation strategies with respect to the naming and formulating are identified and mechanisms are proposed to make DE mutation strategies more consistent.

1.5 Organization of the thesis

Chapter 1 describes problem statement, motivation & background and contribution to current literature. Chapter 2 contains basics concepts of evolutionary algorithms, basic terminologies and evolutionary algorithms evaluation criteria. Chapter 3 describes DE algorithm, its operators and brief survey of control parameters based enhancements of DE algorithm. Chapter 4 presents literature Conflict identification and proposed correction of these inconsistencies. In Chapter 5 we present proposed tournament selected based variants of DE algorithm. Chapter 6 presenters another parent selection based improvement in DE algorithm. Chapter 7 presents a pool based enhancement in DE algorithm. Artificial neural network training using proposed enhancements is presented in Chapter 8. Conclusion and future works are given in Chapter 9 of this dissertation. Appendix contains the detail of benchmark functions used in this thesis.

Chapter 2: Evolutionary Computing Algorithms and Function Optimization

2.0 Chapter Summary

This chapter focuses on the basics of evolutionary algorithms including DE algorithm. Fundamental terminologies of function optimization are also defined in this chapter.

2.1 Introduction to Evolutionary algorithms

The theory of evolutionary algorithms is borrowed from the Darwin's theory of evolution that describes the survival of the fittestt through natural selection and the fitness improvement of individual species. Evolution by means of natural selection of randomly selected individuals helps in searching the optimum value in the space of possible chromosome values. Evolutionary computing algorithms are computational algorithms based on the biological evolutionary process of a stochastic search for an optimal solution to a given problem that adapt genetic evolution and survival of the fittest. The new solutions are generated by evolutionary operators like mutation, crossover and selection. Evolutionary computation tries to model the natural evolutionary process for a successful survival battle, where reproduction and fitness play dominant roles [14**]**. Several computer scientists studied evolutionary systems with the idea that evolution could be used as an optimization tool for solving problems. The idea in all these systems is to evolve a population of candidate solutions to a given problem, using operators inspired by natural genetic variation and natural selection [15]. Many computational problems like resource allocation, path planning, traveling salesman, circuit layout, finding optima from a given search space, etc. require searching through a huge number of possibilities for solutions [15]. Figure 2.1 shows the taxonomy of evolutionary computing algorithms given by Weise [16].

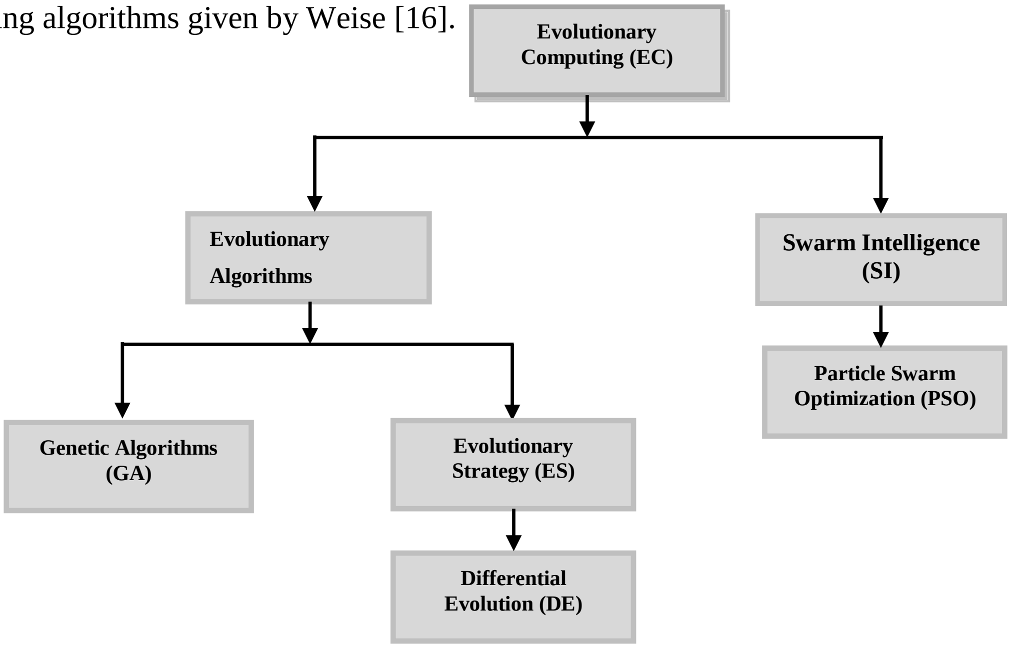


Figure 2.1 Taxonomy of evolutionary computing algorithms **[**16**]**

Evolutionary computation is a computational technology made up of a collection of randomized global search paradigms for finding the optimal solutions to a given problem [14]. Evolutionary computing is an optimization paradigm based on the mechanisms of evolution, such as biological genetics and natural selection [17]. In evolutionary computing we model a population of individuals, where an individual is referred to as a chromosome. A chromosome defines the characteristics of individuals in the population. Each characteristic is called a gene, the value of a gene is known as an allele [18]. The evolutionary computing algorithms have proved themselves to be efficient and robust as compared to brute force algorithms specially when there are infinite possible solutions and we want to find the best possible solution(s) among those solutions. The key objective of the optimization models is to get the best possible choice among the candidate solutions [19]. Best choice refers to an acceptable or satisfactory solution that can be absolute best or any of the best solution over a set of candidate solutions. The advantage of evolutionary computing over other types of numerical methods is its ability to escape from local minima [19-21] however, it does not guarantee to find the exact global optima [16,18,19]. Figure 2.2 contains general steps of evolutionary computing algorithms.

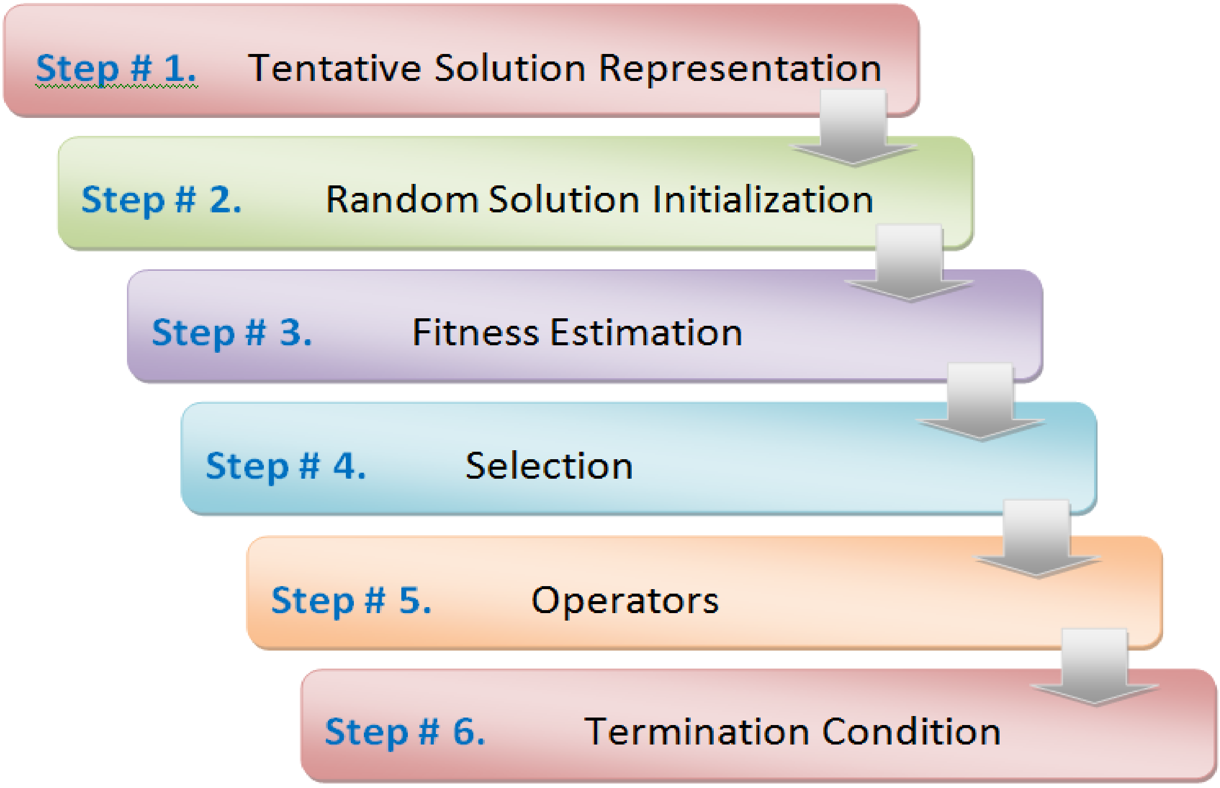


Figure 2.2 Steps of evolutionary algorithms

2.2 Differential Evolution algorithms

Differential Evolution proposed by Storn and Price [22] is a stochastic population based evolutionary algorithm. DE is a simple but powerful technique for optimization problems over continuous spaces. A small number of control parameters makes DE algorithm suitable to apply for optimization tasks. The advantage of DE over other evolutionary algorithms is that it is simple, easy to use, speedy and possess greater probability of finding the global optima for function optimization[18,23-26]. Another advantage of DE is that it has ability to escapes from local minima [27]. DE has been applied to several engineering design problems both as single objective and multi-objective optimization techniques **[**28**]**. Population of potential solutions of DE algorithm is randomly initialized within an n-dimensional search space where all potential solutions are equally likely to be selected as a parent. The candidate solutions evolve themselves by exploring the entire search space overtime to locate the optima of the objective function [**29**]. In this section individual representation, operators, pseudocode and flowchart are discussed to clarify the concept of DE algorithm.

2.2.1 DE Individual Representation

An individual in DE population is a D-dimensional parameter vector that represents a candidate solution of the underlying problem. The initial population is created randomly that is supposed to cover the entire search space. Each vector in the DE is represented by $x_{i,G}$ where i=1,2,3,.,NP; NP is population size and G is generation number.

2.2.2 DE Operators

New population members in DE algorithm are created using various operators like mutation, crossover and selection given in figure 2.3.

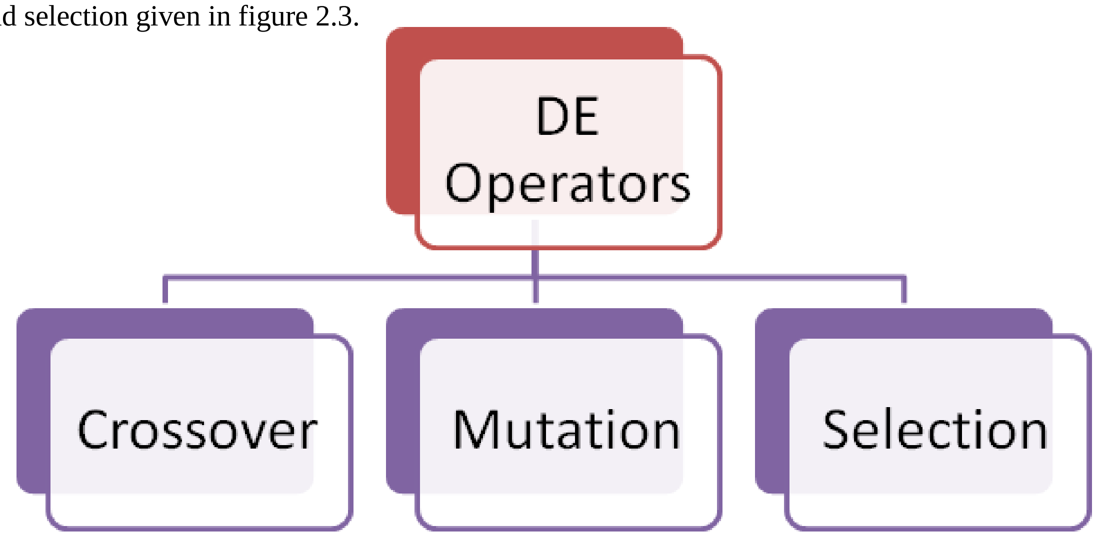


Figure 2.3: DE Operators

(a) Mutation

In mutation operation, mutant vector or donor vector is created that is used to form a trial vector in DE algorithm. Donor vector can be calculated by adding the weighted difference of two vectors ($x_{r_2,G}$, $x_{r_3,G}$) to a third vector ($x_{r_1,G}$).

$$v_{i,G} = x_{r_1,G} + F(x_{r_2,G} - x_{r_3,G}) \quad \text{...(1)}$$

where random indices $r_1, r_2, r_3 \in \{1,2,3,........NP\}$ and $i \neq r1 \neq r2 \neq r3$ and F is the mutation probability parameter.

(b) Crossover

During crossover operation, trial vector is created. Trial vector is created using the donor vector and target vector in DE algorithm using equation (2). The trial vector $u_{i,G+1}$ for jth individual is formed by the following equation.

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & if\ (rand[0,1] \leq CR\ or\ \ j = rnbr[1,D]) \\ x_{ji,G} & if\ (rand[0,1] > CR\ and\ \ j \neq rnbr[1,D]) \end{cases} \quad \text{....................................(2)}$$

where $j = 1,2,3.....,D$ and $rnbr[1,D]$ will return one random generated integer in the specified range. The rand function will generate a random number within the specified interval, D is the dimension and CR is a crossover rate parameter.

(c) Selection

In DE algorithm new population members are formed using selection operation. The selection operator uses the greedy approach by comparing fitness of trial vector $u_{i,G+1}$ with the fitness of target vector $x_{i,G}$; the vector having best fitness is selected as a member of new population. For selection following equation is used

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & if\ (\ fitness(u_{i,G+1}) < fitness(x_{i,G}))) \\ x_{i,G} & otherwise \end{cases} \quad \text{...............(3)}$$

where fitness ( ) function calculates the fitness value from of objective function.

2.2.3 DE Parameters and variables

DE algorithm has three main parameters, mutation probability (F), crossover rate (CR) and population size (NP) [30-31]. New information into the population is incorporated by using mutation operator of DE algorithm while crossover operator exchange information between trial vector and target vector [32]. DE algorithm is sensitive in the selection of its parameter values [33-35]. The control parameter F increases the convergence where Small value of F focuses on exploitation and large value of F enhances exploration ability [33]. Crossover scheme furnishes potential decomposability in the population by shuffling competing vectors[36-37].The population size (NP) represents the number of individual used in searching the optimal value.

2.4 DE Algorithm Flow Chart and Pseudocode

Figure 2.6 shows the overall flowchart that how DE algorithm can be implemented. This flowchart shows various stages where the whole process starts with the random population initialization and ends at optimal solution.

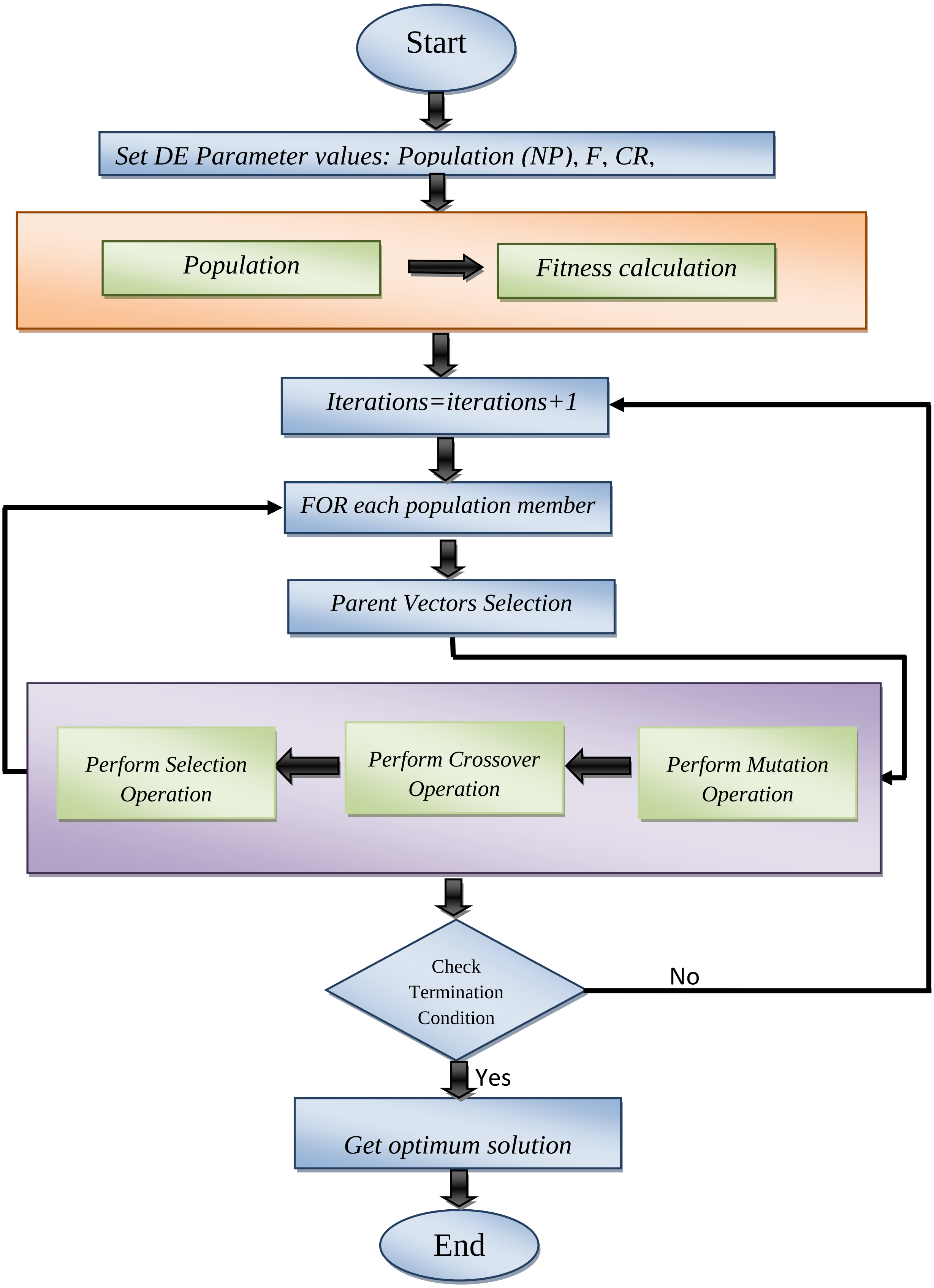


Figure 2.4: DE Algorithm Flowchart

Figure 2.7 shows the pseudocode of DE algorithm. This pseudocode contains an implementation flow of DE algorithm that starts with the random initialization of DE population members. After initialization the fitness value of each population member is calculated then DE population is evolved by applying DE operators mutation, crossover and selection. After evolving the population the optimal solution is obtained.

---

1. Generate the initial population $P_G = \{X_{1,G}, \ldots\ldots X_{NP,G}\}$ for generation *G*=0 and randomly initialize each population member $X_{i,G} = \{x_{i,G}^1 \ldots\ldots\ldots x_{i,G}^D\}$ where $i = 1 \ldots\ldots\ldots\ldots NP$ (population size)

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

    FOR *i* = 1 to NP

    For the i[th] target vector $X_{i,G}$ generate a donor vector $V_{i,G} = \{v_{i,G}^1 \ldots\ldots v_{i,G}^D\}$ with the specified mutation strategy (Standard DE generated mutant vector using equation-I)

    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 $U_{i,G} = \{u_{i,G}^1 \ldots\ldots u_{i,G}^D\}$ 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 e $X_{i,G}$ with fitness function *f*

    IF $f(U_{i,G} \leq f\left(X_{i,G}\right)$, THEN $X_{i,G+1} = U_{i,G}$, $f(X_{i,G}) = f\left(U_{i,G}\right)$

    IF $f(U_{i,G} \leq f\left(X_{best,G}\right)$, THEN $X_{best,G+1} = U_{i,G}$,

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

    END IF

    END IF

    END FOR

    Step 3.4 increment generation number G=G+1

    END WHILE

4. *Get Optimum individual*

Figure 2.5 *Pseudocode of Differential evolution algorithm (DE)*

2.5 DE crossover variants

DE crossover strategies control the number of inherited components of the mutant vector to form a target vector. Binomial and Exponential are main crossover schemes [25,38-40]. The DE crossover rate parameter (CR) influences the size of the perturbation of the base (target) vector to ensure the population diversity [41-42]. Following are the binomial and exponential crossover schemes.

2.5.1 Binomial Crossover

In the crossover operation of DE algorithm a trial vector is formed. In binomial crossover scheme the trail vector $u_{i,G} = u_{i,1,g}, u_{i,2,g}, ..., u_{i,D,g}$ is generated by the following equation

$$u_g^i = \begin{cases} v_{i,j,g} & for\ j = \langle l \rangle_D + \langle l+1 \rangle_D + ...+ \langle l+L-1 \rangle_D \\ x_{i,j} & otherwise \end{cases}$$

where j_{rand} is a randomly chosen integer in the range [1, D], $rand_j(0,1)$ is a random number in (0, 1), $v_{i,j,G}$ is the donor vector. CR is the crossover control parameter in the range. Due to the range of j_{rand}, $u_{i,G}$ is different from $x_{i,j}$ and index $i = 1,2,3,..........., NP$ and $j = 1,2,3,.........., D$.

The C-Style pseudocode of binomial version of *"DE/rand/1/bin"* is

```
r==rand(0,D)
for(k = 0; k < DIM ; k + +)
{
        if (Rand_double(0; 1) < CR|| k == DIM-1)
          {
                trial[r]= pvector[a].x[r]+F*( pvector[b].x[r]- pvector[c].x[r]);
          }
        else
                trial[r]=pvector[index].x[r];
                r=(r+1)%DIM;
}
```

Example of binomial crossover for donor vector and target vectors is as follows

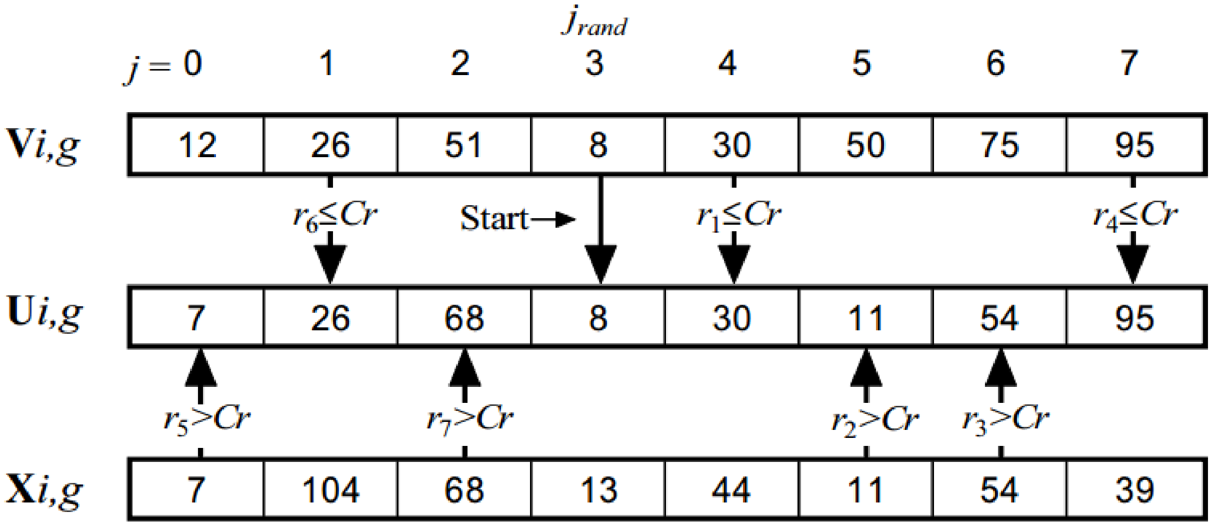


Figure 2.6: Binomial Crossover Example **[**26**]**

It starts from any random location say j_{rand} by generating a random number for each dimension if that random number is less than the CR then takes the current dimension of trial vector parameters are inherited from the mutant vector otherwise inherited from the target vector. The condition $k == rand(0, D)$ is used to ensure that at least one dimension in the trail vector should be from mutated population member to incorporate the diversity in the new population member.

2.5.2 Exponential Crossover

In the exponential crossover scheme the trail $u_{i,G} = u_{i,1,g}, u_{i,2,g}, ..., u_{i,D,g}$ is created as follows

$$u_g^i = \begin{cases} v_{i,j,g} & if \left(rand_j \leq CR \, or \; j = j_{rand} \right) \\ x_{i,j} & otherwise \end{cases}$$

i = 1, 2, 3,….,NP , j = 1, 2, 3,…..,D and < >D denotes the modulo function with modulus D. The starting index l is chosen at random from [1, D]. L is also a randomly generated number from [1, D]. The parameters l and L are regenerated for each trial vector $U_{i,G}$.

Example of exponential crossover for donor vector and target vectors is as follows.

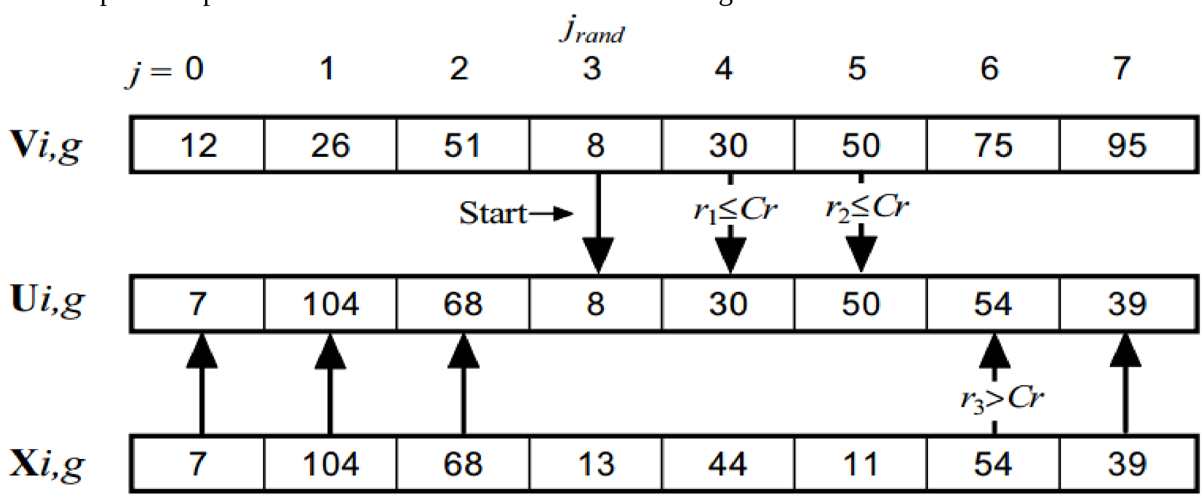


Figure 2.7: Exponential Crossover Example **[**26**]**

The C-Style pseudocode of exponential version of *"DE/rand/1/exp"* is

```
r = rand(0; DIM )
for(i = 0; i < DIM ; i + +)
{
        trial_vector[i] = pvector[index].x[i];
}
j=r;
l=0;
do
{
        trial_vector = pvector[a].x[j ] + F ( pvector[b].x[j ] - pvector[c].x[j ]);
        j = (j + 1)%DIM ;
        l=l+1;
}
```

It starts from any random location say j_{rand}. It mutates each dimension of trial vector until a random number greater than CR or we reach at the maximum dimension of the current individual. This crossover scheme incorporates less diversity in the new individual as binomial crossover scheme since it terminates if a single test condition is false.

2.6 Function Optimization

Function optimization is the process of finding most suitable value for a function within a given domain. For a function f(x), called the objective function, that has a domain of real numbers of set A, the maximum optimal solution occurs where $f(x_0) \leq f(x_1)$ over set A and the minimum optimal solution occurs where $f(x_0) \geq f(x_1)$ over set A. The important terms used in function optimization are discussed as follows:

2.6.1 Fitness/objective Function

It is the measurement of the anticipated function of an individual in the population. The fitness function, f, maps a chromosome representation into a scalar value $F : X^{n_x} \to R$ where X represents the data type of the chromosome.

2.6.2 Monotonic Function

A function $f(x)$ for any points x_1, x_2 with $f(x_1) \leq f(x_2)$ is said to be monotonic increasing and is monotonic decreasing if $f(x_1) \geq f(x_2)$.

2.6.3 Local & Global Minimum

(a) Local minimum

Local minimum is further divided into local and global optima

(i) Strong local minimum

The solution $x^* \in F$ is a strong local minimum of f if $x_N^* < f(x), \forall x \in N$ where $F \subseteq N$ is a set of feasible points in the neighborhood of x_N^*.

(ii) Weak local minimum

The solution $x^* \in F$ is a strong local minimum of f if $x_N^* \leq f(x), \forall x \in N$ where $F \subseteq N$ is a set of feasible points in the neighborhood of x_N^*.

(b) Global Minimum

The solution $x^* \in F$ is a global optimum of the objective function f if $f(x^*) < f(x), \forall x \in F$ where $F \subseteq S$ and S is search space of candidate solutions. The graph of strong local optima, weak local minima and global optima is given in figure 2.8

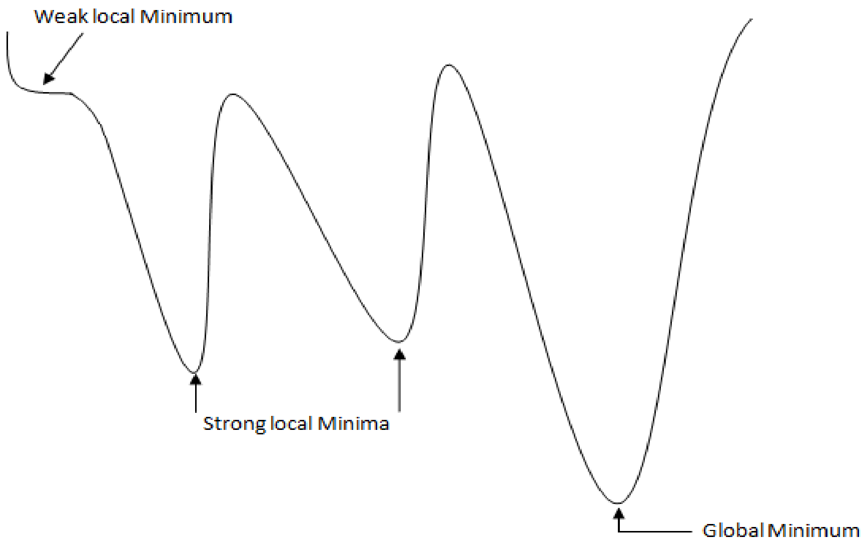


Figure 2.8: Global and local minima **[**19**]**

2.6.5 Local search

Assume that f is a measurable function, S is a measurable subset of IR^{N_x} and both the algorithm and the convergence conditions are satisfied. Let $Lim_{t=0}^{\infty}(X_t)$ be a sequence generated by the algorithm A, then $\lim_{t\to 0} P(X_t \in R_s) = 1$ where $P(X_t \in R_s)$ is the probability that at step t, the point X_t generated by the algorithm is in the optimality region R_s.

2.6.6 Global Search

Assume that f is a measureable function, S is a measureable subset of IR^{N_x} and both the algorithm condition and convergence condition for global search are satisfied. If $Lim_{t=0}^{\infty}(X_t)$ is a sequence generated by the algorithm A then $\lim_{t\to 0} P(X_t \in R_s) = 1$ where $P(X_t \in R_s)$ is the probability that the point X_t generated by A is in R_s at time step t.

2.6.7 Unimodal

A function $f(x)$ is unimodal in $a \le x \le b$ iff it is monotonic on either side of x^+ where x^+ is the optimal point.

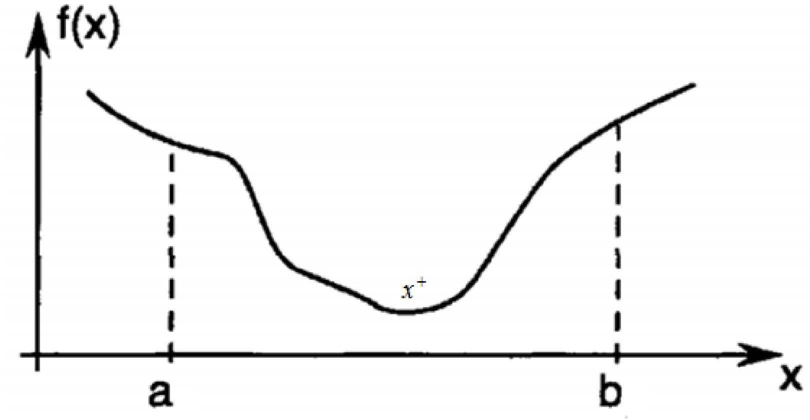


Figure 2.9: Unimodal **[**43**]**

2.6.8 Multimodal

A mathematical function having multiple (local) optima is called multimodal function.

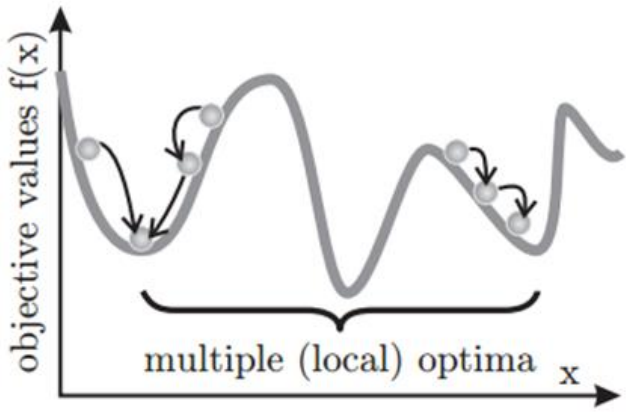


Figure 2.10: Multiple Local Optima **[**43**]**

2.6.9 Separable

A function $f(x)$ of n variables is separable if it can be rewritten as a sum of n functions of just one variable. The x_i parameters of a separable function are independent of each other.

$$\arg\min(x_1, x_2, \ldots.., x_n)\, f(x_1, x_2, \ldots., x_n) = \arg\min_{(x_1)} f(x_1, \ldots\ldots.), \ldots\ldots, \arg\min_{(x_n)} f(\ldots\ldots., x_n)$$

2.6.10 Non-Separable

A function $f(x)$ is called m-nonseparable function if at most m of its parameters is not independent. A nonseparable function is called fully-nonseparable function if any two of its parameters are not independent. Fully-nonseparable function is considered to be most difficult problems.

2.7 Chapter Summary

This chapter presented details about DE algorithm. Implementation of DE algorithm in the form of flowchart and pseudocode is shown in this chapter. Various parameters and terms related to DE algorithm and function optimization such as crossover, mutation, selection, local optima, global optima, separable and non separable are also discussed.

Chapter # 3: DE Algorithm in the light of literature

3.0 Chapter Summary

This chapter presents literature review on the important aspects of DE algorithm and related topics. This is an important chapter where major limitations of the DE algorithm are also highlighted. The development in terms of parameters, mutation strategies and current applications are also discussed in this chapter.

3.1 DE Mutation strategies

There are several DE algorithm mutation strategies that are formed by the linear combination of existing population members. The trial/mutant vector and target vector forms the mutant vector in DE. Throughout this thesis 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 strategies are used to create trial vector by using current, best, better and random vectors. The behaviour of DE algorithm is influenced by the selection of mutation strategy and crossover scheme along with their control parameters: mutation probability 'F' and Crossover rate 'CR' [25, 40].

Difference Vector (DV) is a difference of two vectors that is used to form offspring in the population **[**44**]**. To form the mutant vector in DE some researcher uses a random value $\lambda \in (0,1)$ as a coefficient multiplier with the first difference vector and mutation probability "F" as a coefficient multiplier with the other difference vector(s) [22,45-47]. Some researchers have used only "F" as a coefficient multiplier with the difference vector(s) to form the mutant vector [25, 48]. To reduce the number of control parameters of DE algorithm we use λ=F [49-53].

Literature shows that DE mutation strategies reveals several inconsistencies in naming and formulation of DE mutation strategies. The detailed list of mutation strategies in terms of binomial and exponential crossover are reported in table-3.1.

Table 3.1: List of DE mutation strategies available in literature

S. No	Binomial/ Exponential		S. No	Binomial/ Exponential	
	Variant/strategy Name	**Equation**		**Variant/strategy Name**	**Equation**
1	DE/rand/1	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3})$	14	DE/current to best/2	$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2})$
2	DE/best/1	$v_g^i = x_g^{best} + F(x_g^{r1} - x_g^{r2})$	15	DE/current to best/2	$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4})$
3	DE/rand/2	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$	16	DE/current to rand/2	$v_g^i = x_g^i + F(x_g^{r1} - x_g^i) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$
4	DE/best/2	$v_g^i = x_g^{best} + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4})$	17	DE/rand to best/2	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4})$
5	DE/current to rand/1	$v_g^i = x_g^i + F(x_g^{r1} - x_g^i) + F(x_g^{r2} - x_g^{r3})$	18	DE/rand to best/2	$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2})$
6	DE/Current-to-rand/1	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^i) + F(x_g^{r1} - x_g^{r3})$	19	DE/rand to best/2	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^i) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$
7	DE/current to best/1	$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2})$	20	DE/rand to best/2	$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4})$
8	DE/current to best/1	$v_g^i = x_g^i + F(x_g^{best} - x_g^{r1}) + F(x_g^{r1} - x_g^{r2})$	21	DE/rand to best/2	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r1}) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$
9	DE/rand to best/1	$v_g^i = x_g^i + F(x_g^{best} - x_g^{r1}) + F(x_g^{r1} - x_g^{r2})$	22	DE/rand to current /2	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^i) + F(x_g^{r3} - x_g^{r4})$
10	DE/rand to best/1	$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2})$	23	DE/rand to best and current /2	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r2}) + F(x_g^{r3} - x_g^i)$
11	DE/rand to best/1	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4})$	24	DE/mid-to-better/1	$v_g^i = F(x_g^{better} + x_g^i)/2 + F(x_g^{better} - x_g^i) + F(x_g^{r1} - x_g^{r2})$
12	DE/rand to best/1	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r1}) + F(x_g^{r2} - x_g^{r3})$	25	DE/rand/3	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5}) + F(x_g^{r6} - x_g^{r7})$
13	DE/rand to best/1	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^i) + F(x_g^{r2} - x_g^{r3})$	26	DE/best/3	$v_g^i = x_g^{best} + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4}) + F(x_g^{r5} - x_g^{r6})$

The list of DE mutation strategies is presented in table 3.1. The detail of formulation and graphical representation of these mutation strategies is discussed as follows

- ***"DE/rand/1"***

"DE/rand/1" is introduced by Storn & Price **[**22**]**. This strategy is known as the basic strategy in the DE algorithm. *DE/rand/1* is used as a default mutation strategy in the standard DE algorithm. This mutation strategy has no conflicts in the literature with other mutation strategies. The equation of *"DE/rand/1"* is

$$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3}) \quad \text{..(3.1)}$$

"DE/rand/1" uses three random vectors x_g^{r1}, x_g^{r2} and x_g^{r3} and one difference vector to generate the mutant vector. The graphical representation of this mutation strategy is given in figure 3.1.

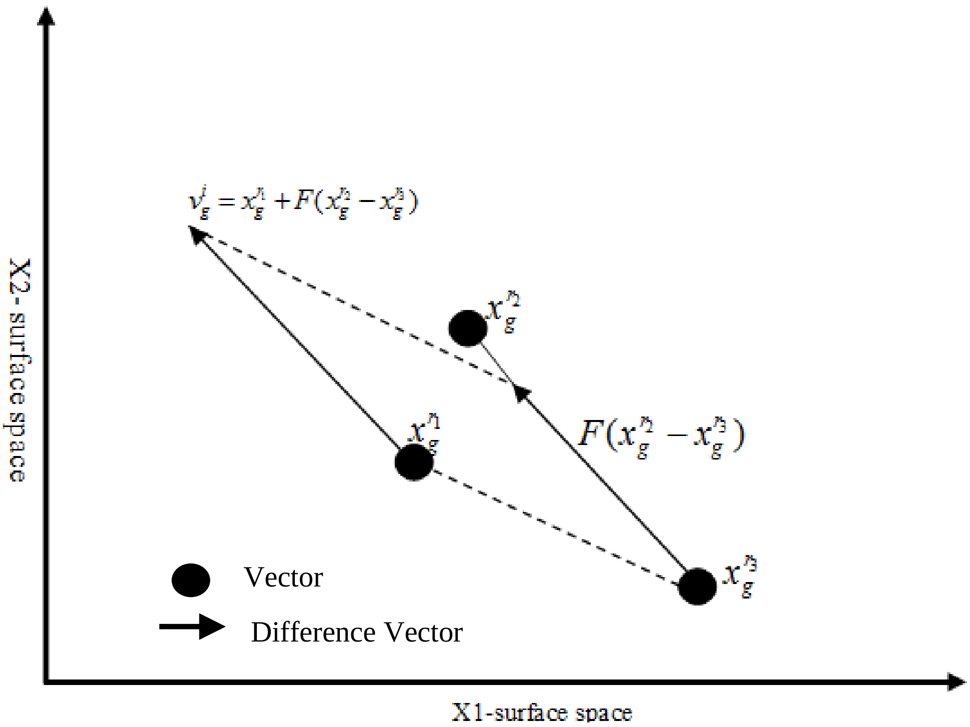


Figure.3.1 *DE/rand/1*

- ***"DE/best/1"***

"DE/best/1" is introduced by Storn **[**46**]**. The author has used it for function optimization application. This mutation strategy has no conflicts with other mutation strategies in literature. The equation of *"DE/best/1"* is

$$v_g^i = x_g^{best} + F(x_g^{r1} - x_g^{r2}) \quad \text{..(3.2)}$$

"DE/best/1/bin" contains two random vectors x_g^{r1}, x_g^{r2} in the difference vector and one best vector x_g^{best} to generate the mutant vector. This mutation strategy utilizes best vector along with difference vector moving from one random vector to another vector. The graphical representation of this mutation strategy is given in figure 3.2.

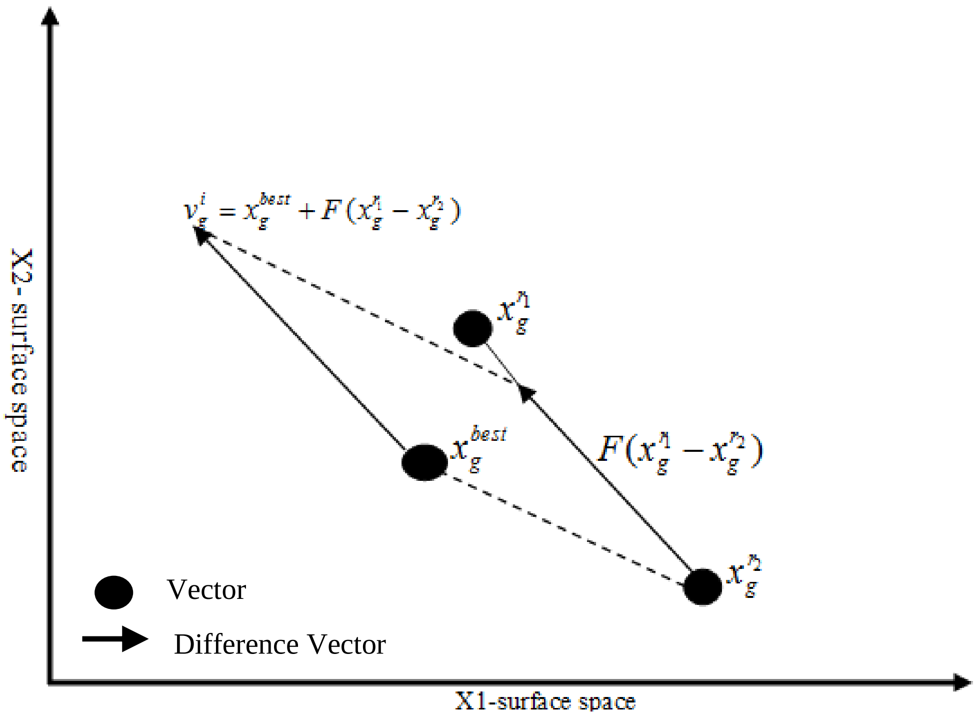


Figure.3.2 *DE/best/1*

- ***"DE/rand/2"***

  Storn & Price [25] introduced *"DE/rand/2"* and used it for function optimization application. This mutation strategy has no conflicts with other mutation strategies in literature. The equation of "*DE/rand/2" is*

  $$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5}) \quad (3.3)$$

  "*DE/rand/2"* uses five random vectors x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4} and x_g^{r5} that perturbs x_g^{r1} with two difference vectors to generate the mutant vector. This mutation strategy generates a mutant vector that is not biased towards best any particular vector. The graphical representation of this mutation strategy is given in figure 3.3.

- ***"DE/best/2"***

Price [54] introduced *"DE/best/2"* mutation strategy and used it for function optimization problems. This mutation strategy has no conflicts with other mutation strategies in literature. The equation of *"DE/best/2"is*

$$v_g^i = x_g^{best} + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4}) \quad (3.4)$$

"DE/best/2" uses four random vectors x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4} and x_g^{best} that perturbs x_g^{best} with two difference vectors without repeating any vector. The graphical representation of this mutation strategy is given in figure 3.4.

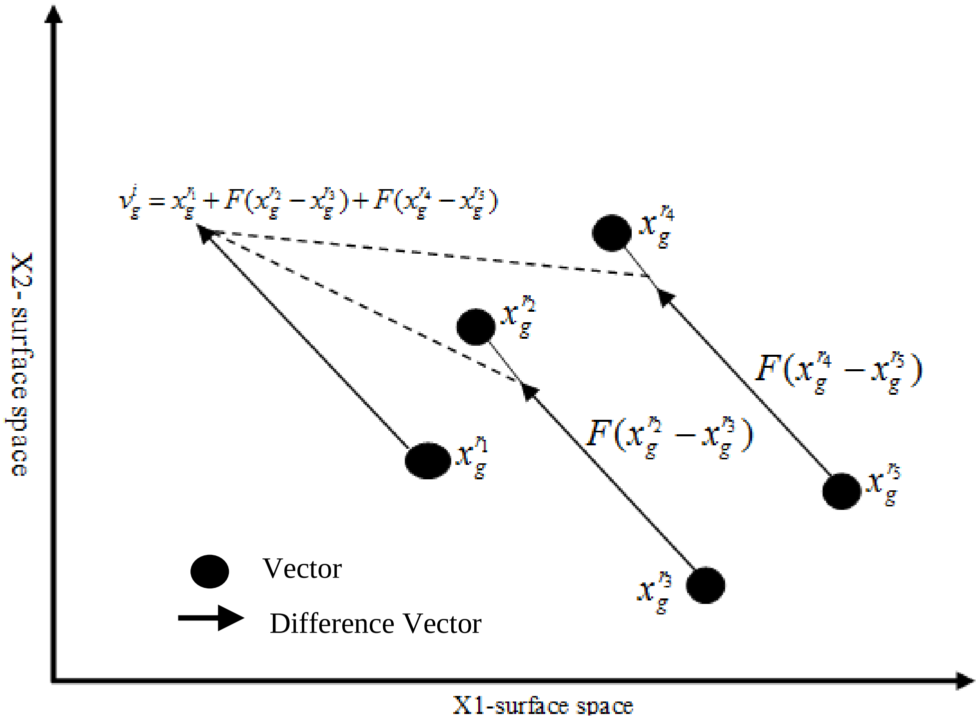


Figure.3.3 *DE/rand/2*

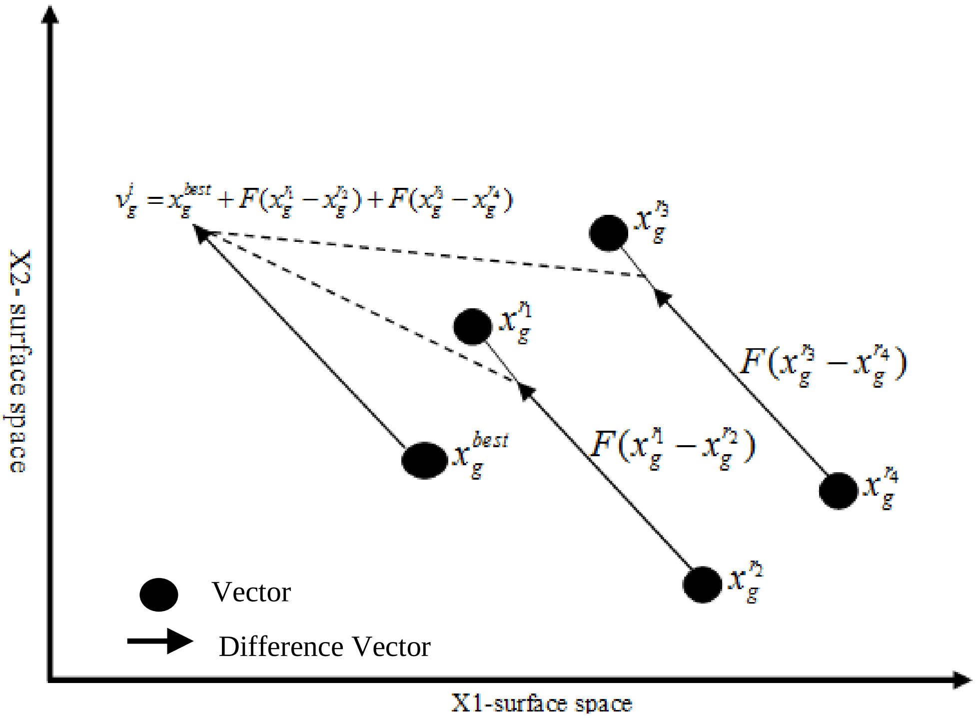


Figure.3.4 *DE/best/2*

- ***"DE/current to rand/1"***

Zielinski et. al [49], Gong & Cai [55], Farsangi & Nezamabadi-pour [56], Gong et.al [57], Jeyakumar & Velayutham [58] have used *"DE/current to rand/1"* mutation strategy in their research work. This mutation strategy has naming conflict (Discussed in chapter 4) with other mutation strategies in the literature. The equation of *"DE/current to rand/1"* is

$$v_g^i = x_g^i + F(x_g^{r1} - x_g^i) + F(x_g^{r2} - x_g^{r3}) \qquad \text{.......................................(3.5)}$$

"DE/current to rand/1" uses three random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}$ and a current vector x_g^i that perturbs x_g^i and places the perturbation between x_g^{r1} and x_g^i with one difference vector utilizing two random numbers. The graphical representation of this mutation strategy is given in figure 3.5.

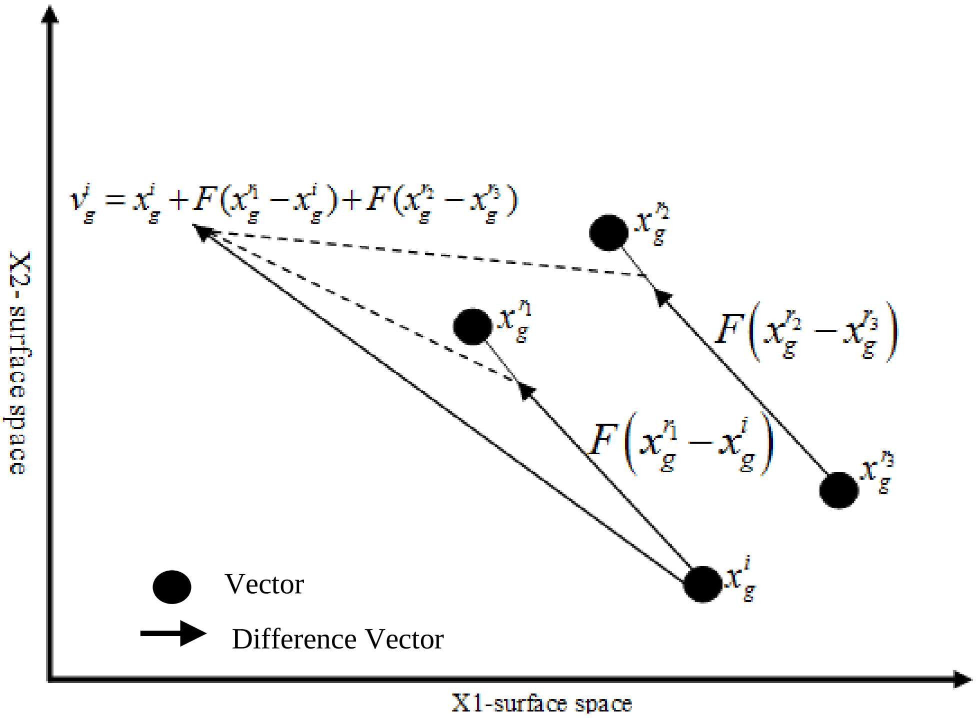


Figure.3.5 *DE/current to rand/1*

- ***"DE/Current-to-rand/1"***

"DE/Current-to-rand/1" mutation strategy is used by Huang et. al **[**59**]** in their research work for constrained real parameter optimization. This mutation strategy has a naming conflict with other mutation strategies in the literature (Discussed in chapter 4). The equation of *"DE/current-to-rand/1"* is

$$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^i) + F(x_g^{r1} - x_g^{r3}) \qquad \text{...(3.6)}$$

"DE/current-to-rand/1" contains three random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}$, a current vector x_g^i in moving from current vector to random vector and one difference vector of two random numbers by repeating a random number (x_g^{r1}).The graphical representation of this mutation strategy is given in figure 3.6.

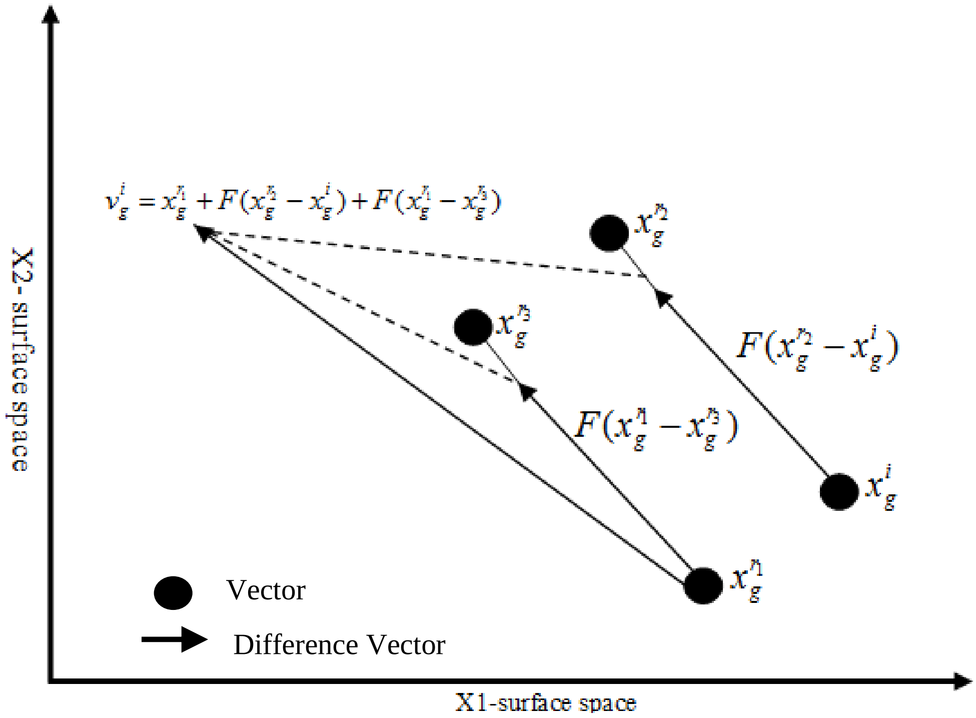


Figure.3.6 *DE/Current-to-rand/1*

- ***"DE/current to best/1"***

Storn & Price **[**22**]** introduced *"DE/current to best/1"* mutation strategy that is used by many researcher [42, 49, 58-77] in their research work. This mutation strategy has a naming and equation conflict (Discussed in chapter 4) with other mutation strategies in the literature. The equation of *"DE/current to best/1" is*

$$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2}) \quad \text{………………………...………...……(3.7)}$$

"DE/current to best/1" contains two random vectors x_g^{r1}, x_g^{r2}, a current vector x_g^i and a best vector x_g^{best} in moving from current vector to best vector and a difference vector of two random numbers. This variant repeats current vector in the first difference vector. The graphical representation of this mutation strategy is given in figure 3.7.

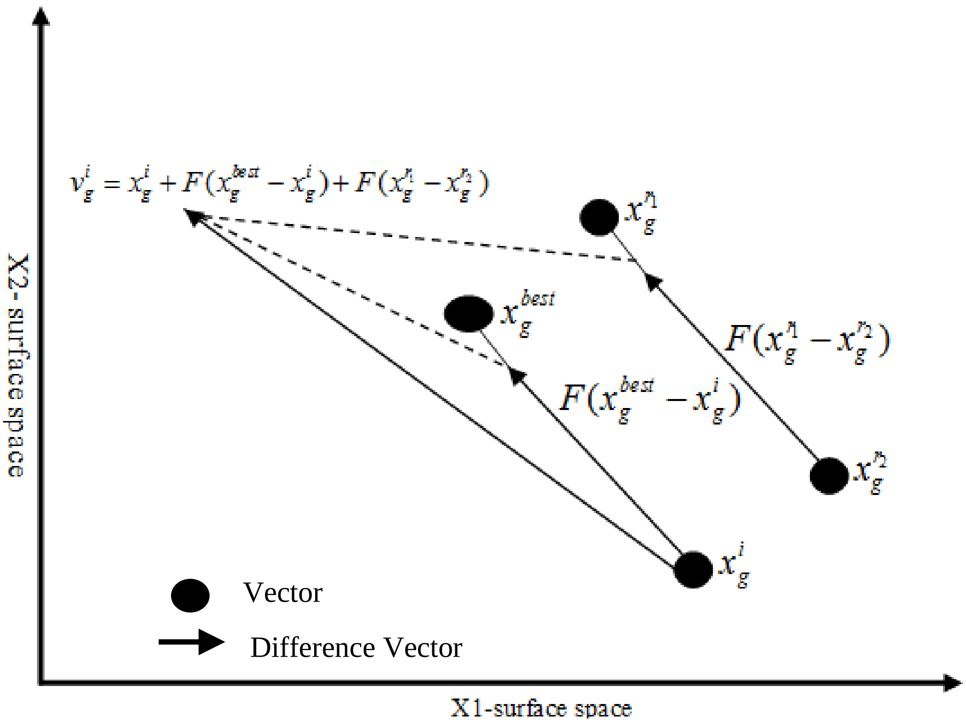


Figure.3.7 *DE/current to best/1*

- ***"DE/current to best/1"***

Podoba et. al **[**78**]** have used *"DE/current to best/1"* mutation strategy in their research of surface reconstruction using AI. This mutation strategy has a naming conflict as well as the equation conflict with other mutation strategies in the literature (Discussed in chapter 4). The equation of *"DE/current to best/1"* is

$$v_g^i = x_g^i + F(x_g^{best} - x_g^{r1}) + F(x_g^{r1} - x_g^{r2}) \qquad \text{.............................(3.8)}$$

"DE/current to best/1" contains two random vectors x_g^{r1}, x_g^{r2}, a best vector x_g^{best} and a current vector x_g^i in moving from random vector to best vector and a difference vector of two random number by repeating random number x_g^{r1}. The graphical representation of this mutation strategy is given in figure 3.8.

- ***"DE/rand to best/1" (given in fig 3.9)***

Davendra *et. al* **[**79**]** have used *"DE/rand to best/1"* mutation strategy for travelling salesman problem. This mutation strategy has a naming and equation conflict with the other mutation strategies in the literature (Discussed in chapter 4). The equation of *"DE/rand to best/1"* is

$$v_g^i = x_g^i + F(x_g^{best} - x_g^{r1}) + F(x_g^{r1} - x_g^{r2}) \qquad \text{.............................(3.9)}$$

"DE/rand to best/1" contains two random vectors x_g^{r1}, x_g^{r2}, a best vector x_g^{best} and a current vector x_g^{i} in moving from random vector to best vector and a difference vector of two random numbers by repeating a random number x_g^{r1}. The graphical representation of this mutation strategy is given in figure 3.9.

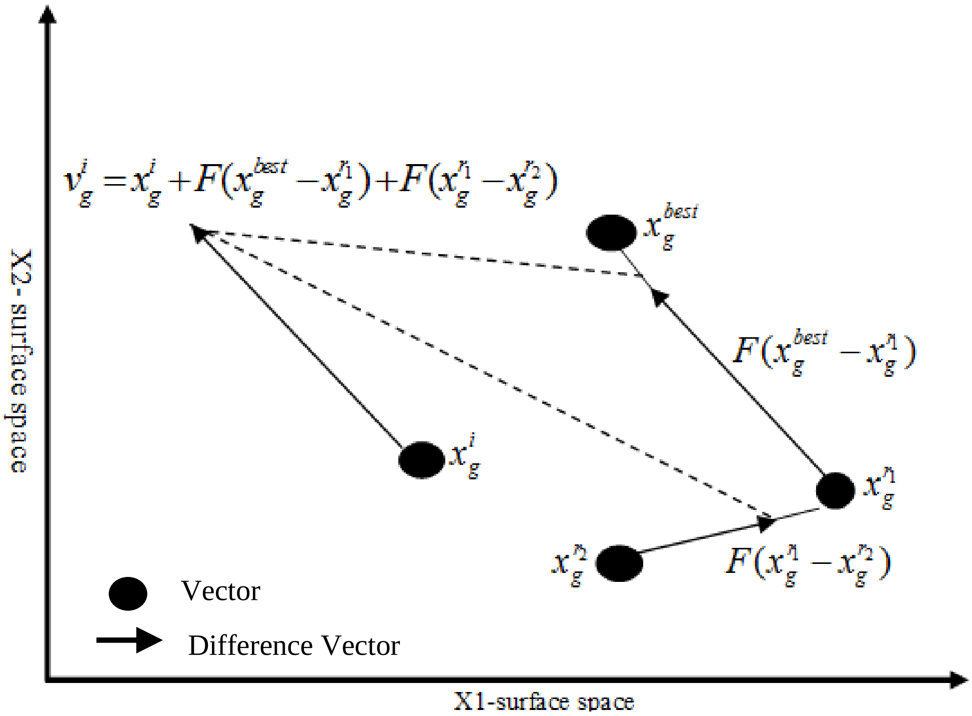


Figure.3.8 *DE/current to best/1*

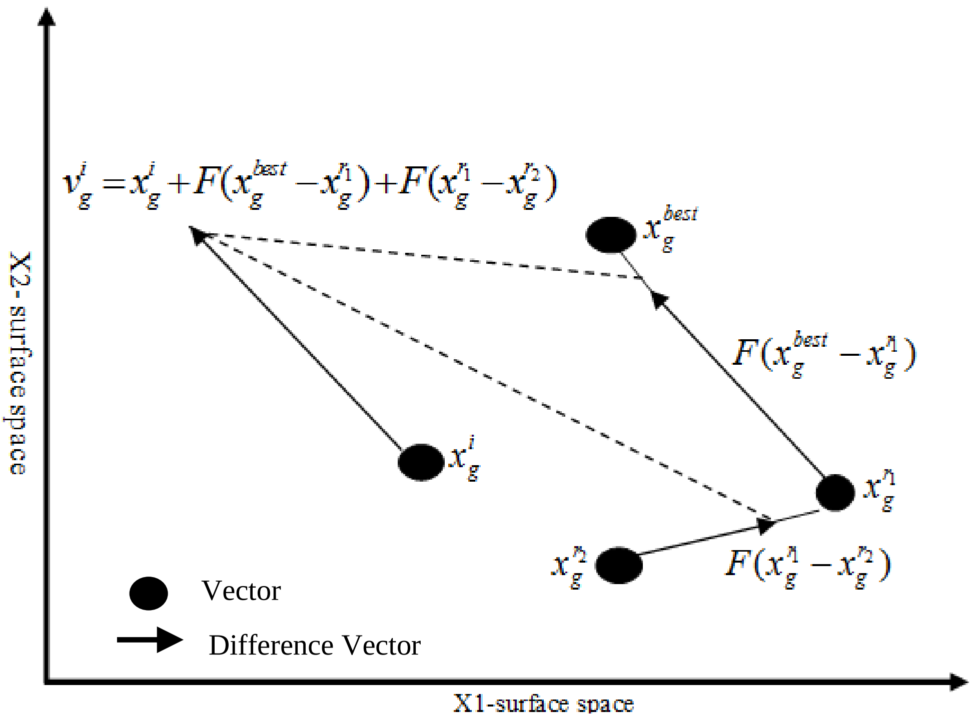


Figure.3.9 *DE/rand to best/1*

- ***“DE/rand to best/1”***

Storn [46] has introduced “*DE/rand to best/1*” and use it for function optimization application. This mutation strategy has been used by many researchers [38, 45, 56, 80-95] in their research work. This mutation strategy has naming conflict as well as equation conflict with other mutation strategies in the literature (Discussed in chapter 4). The equation of “*DE/rand to best/1” is*

$$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2}) \quad \text{...........(3.10)}$$

“*DE/rand to best/1*” contains two random vectors x_g^{r1}, x_g^{r2}, a best vector x_g^{best} and a current vector x_g^i in moving from current vector to best vector and a difference vector of two random numbers. The graphical representation of this mutation strategy is given in figure 3.10.

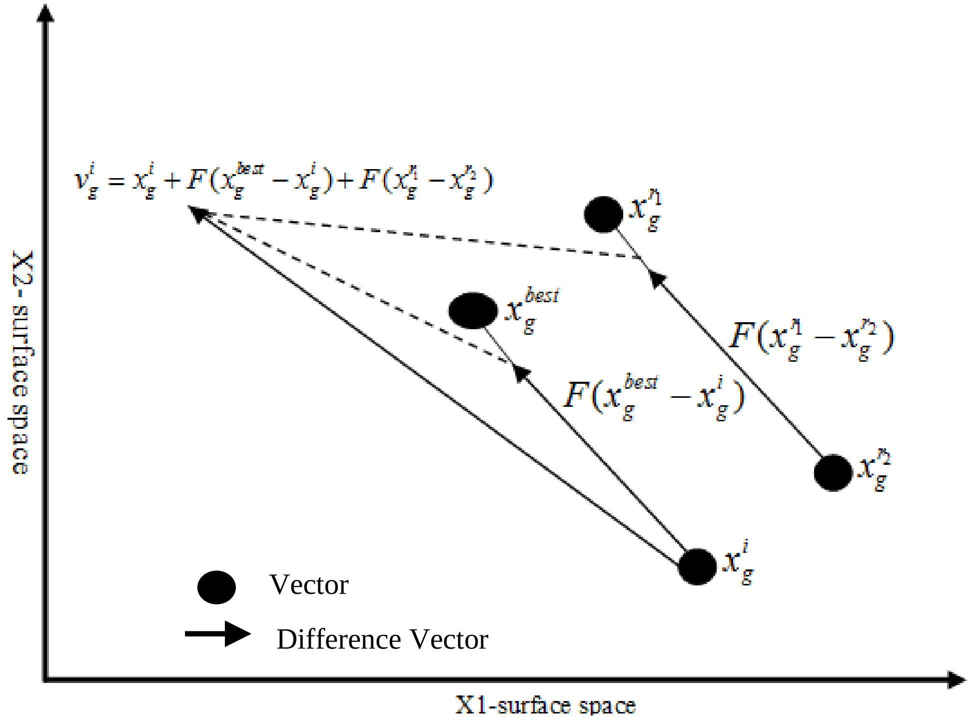


Figure.3.10 *DE/rand to best/1*

- ***“DE/rand to best/1”***

“DE/rand to best/1” mutation strategy is used by Ali et. al [40], Zhao et.al [96] and Pant et.al [97] their research work. This mutation strategy has naming conflict with other mutation strategies in the literature (Discussed in chapter 4). The equation of *“DE/rand to best/1”* is

$$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4}) \quad \text{...........(3.11)}$$

"DE/rand to best/1" contains four random vectors x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4} and a best vector x_g^{best} in moving from a random vector to a best vector and a difference vector of two random numbers. The graphical representation of this mutation strategy is given in figure 3.11.

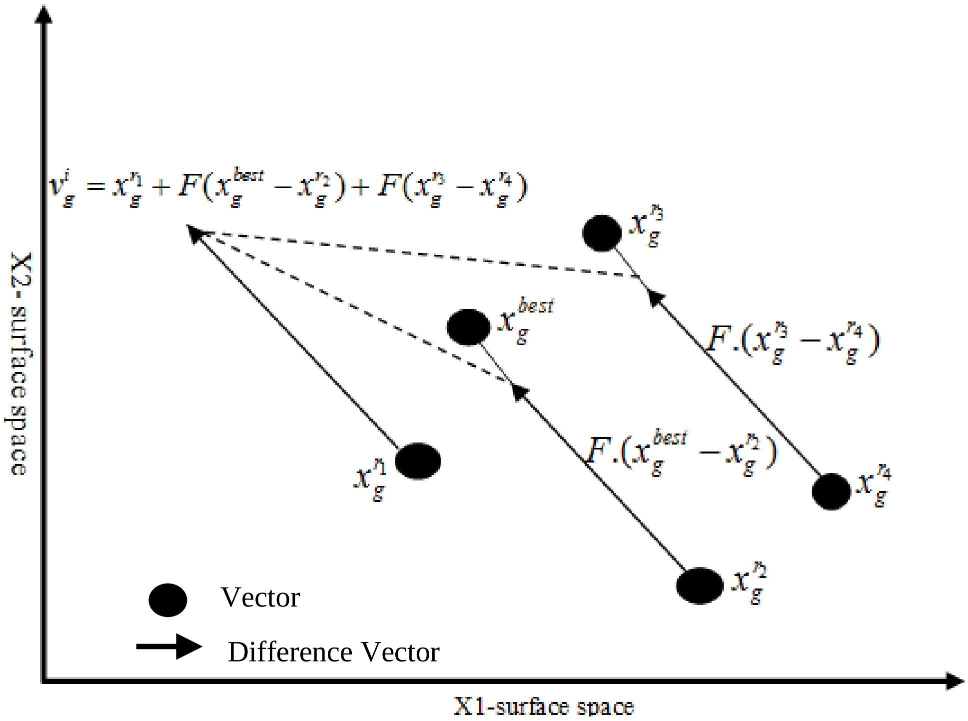


Figure.3.11 *DE/rand to best/1*

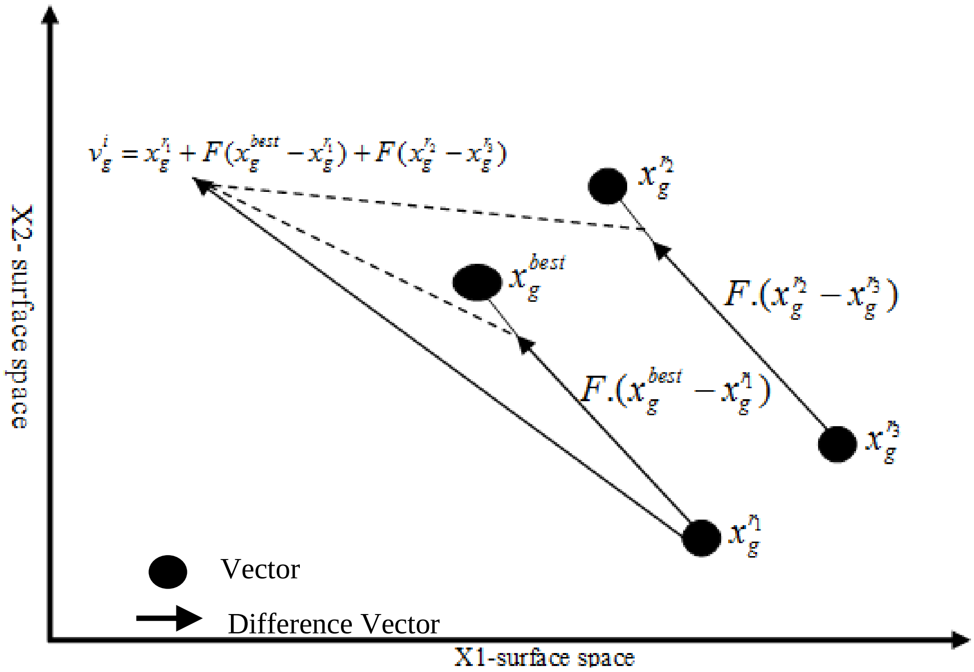


Figure.3.12 *DE/rand to best/1*

- ***"DE/rand to best/1"***

Almeida-Luz et. al **[**98**]**, Mendes & Mohais **[**99**]** have used "*DE/rand to best/1*" mutation strategy in their research work. This mutation strategy has naming conflict with other mutation strategies in the literature (Discussed in chapter 4). The equation of "*DE/rand to best/1*" is

$$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r1}) + F(x_g^{r2} - x_g^{r3}) \qquad \text{...(3.12)}$$

"*DE/rand to best/1*" contains three random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}$ and a best vector x_g^{best} in moving from random vector to best vector and a difference vector of two random vectors. The graphical representation of this mutation strategy is given in figure 3.9. The graphical representation of this mutation strategy is given in figure 3.12.

- ***"DE/rand to best/1"***

Jeyakumar & Velayutham **[**58**]** have used *"DE/rand to best/1"* mutation strategy for empirical analysis of DE variants over function optimization problem. This mutation strategy has naming conflict with other mutation strategies in the literature (Discussed in chapter 4). The equation of *"DE/rand to best/1"* is

$$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{i}) + F(x_g^{r2} - x_g^{r3}) \qquad \text{...(3.13)}$$

"DE/rand to best/1" contains three random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}$, a current vector x_g^i and a best vector x_g^{best} in moving from current vector to best vector and a difference vectors of two random vectors. The graphical representation of this mutation strategy is given in figure 3.13.

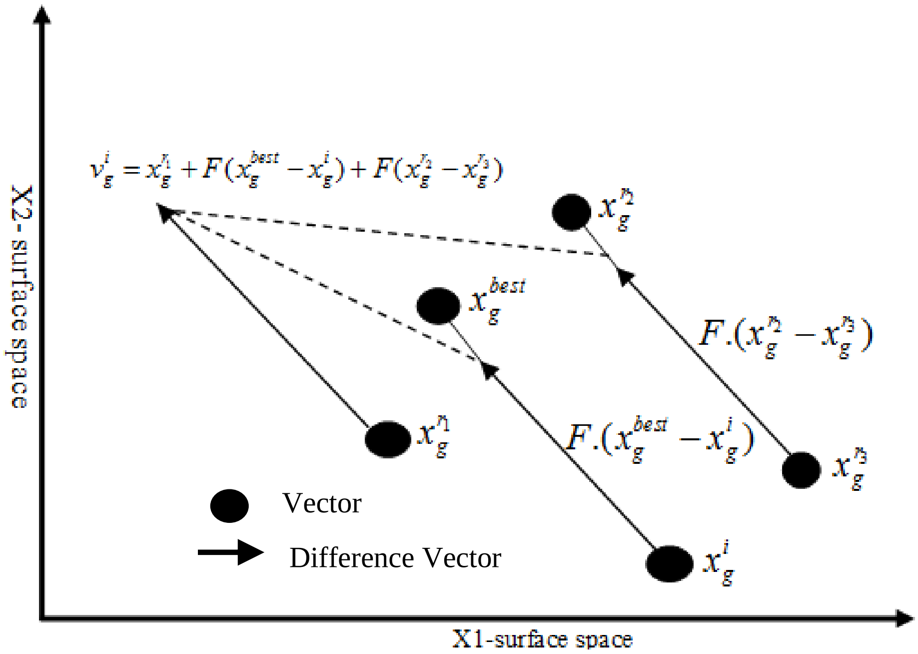


Figure.3.13 *DE/rand to best/1*

- ***"DE/current to best/2"***

"*DE/current to best/2*" mutation strategy is used by many researchers[100-107] in their research work for various problems. This mutation strategy has naming and equation conflicts with other mutation strategies in the literature (Discussed in chapter 4). The equation of "*DE/current to best/2*" is

$$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2}) \quad \text{...(3.14)}$$

"*DE/current to best/2*" contains two random vectors x_g^{r1}, x_g^{r2}, a best vector x_g^{best} and a current vector x_g^i in moving from current vector to a best vector and a difference vector of two random numbers. The graphical representation of this mutation strategy is given in figure 3.14.

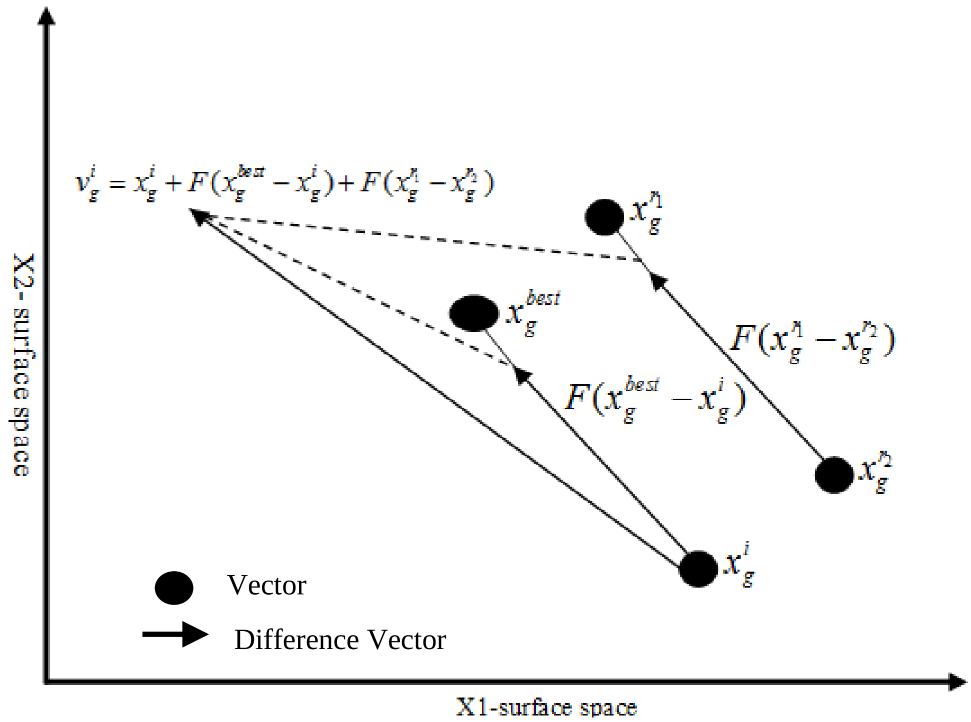


Figure.3.14 *DE/current to best/2*

- ***"DE/current to best/2"***

Zielinski et. al **[**49**]**, Huang et. al **[**59**]** and Epitropakis et. al **[**76**]** have used "*DE/current to best/2*" mutation strategy in their research work. This mutation strategy has naming conflict with other mutation strategies in the literature (Discussed in chapter 4). The equation of "*DE/current to best/2*" is

$$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4}) \quad \text{.................................(3.15)}$$

"*DE/current to best/2*" contains four random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4}$ a best vector x_g^{best} and a current vector x_g^i in moving from current vector to best vector and two difference vectors of random numbers. The graphical representation of this mutation strategy is given in figure 3.15.

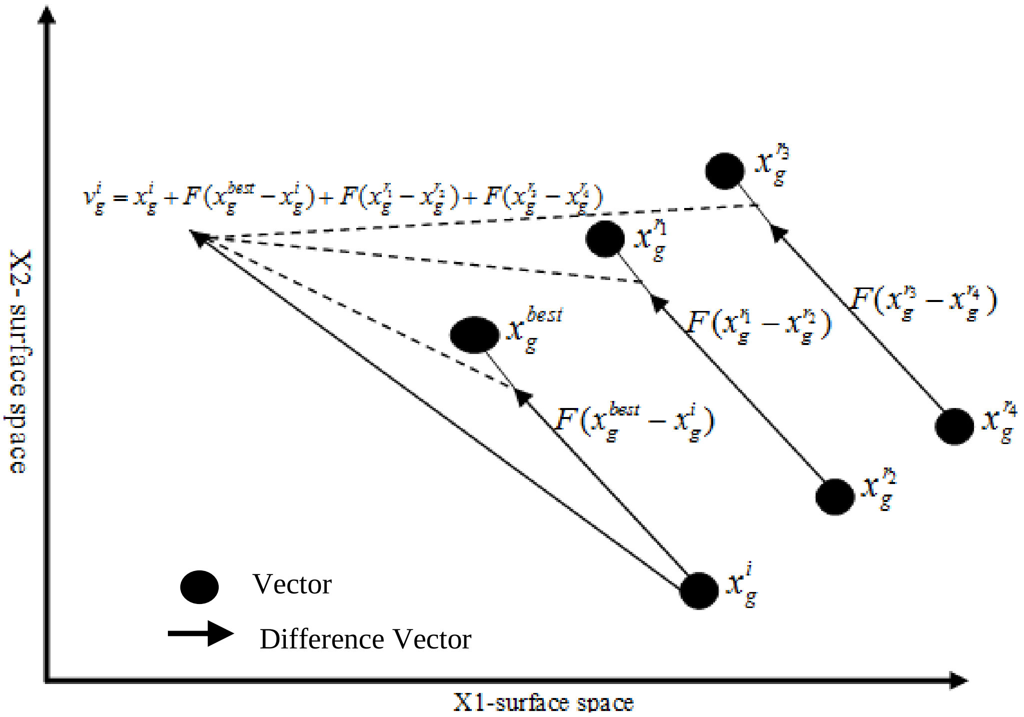


Figure.3.15 *DE/current to best/2*

- ***"DE/current to rand/2"***

Zielinski et. al **[**49**]** have used *"DE/current to rand/2"* mutation strategy in their research work on choosing suitable variants of differential evolution and particle swarm optimization. This mutation strategy has naming conflict with other mutation strategies in the literature (Discussed in chapter 4). The equation of *"DE/current to rand/2"* is

$$v_g^i = x_g^i + F(x_g^{r1} - x_g^i) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5}) \quad \text{.......................................(3.16)}$$

"DE/current to rand/2" contains five random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4}, x_g^{r5}$ and a current vector x_g^i in moving from current vector to a random vector and two difference vectors of random numbers. The graphical representation of this mutation strategy is given in figure 3.16.

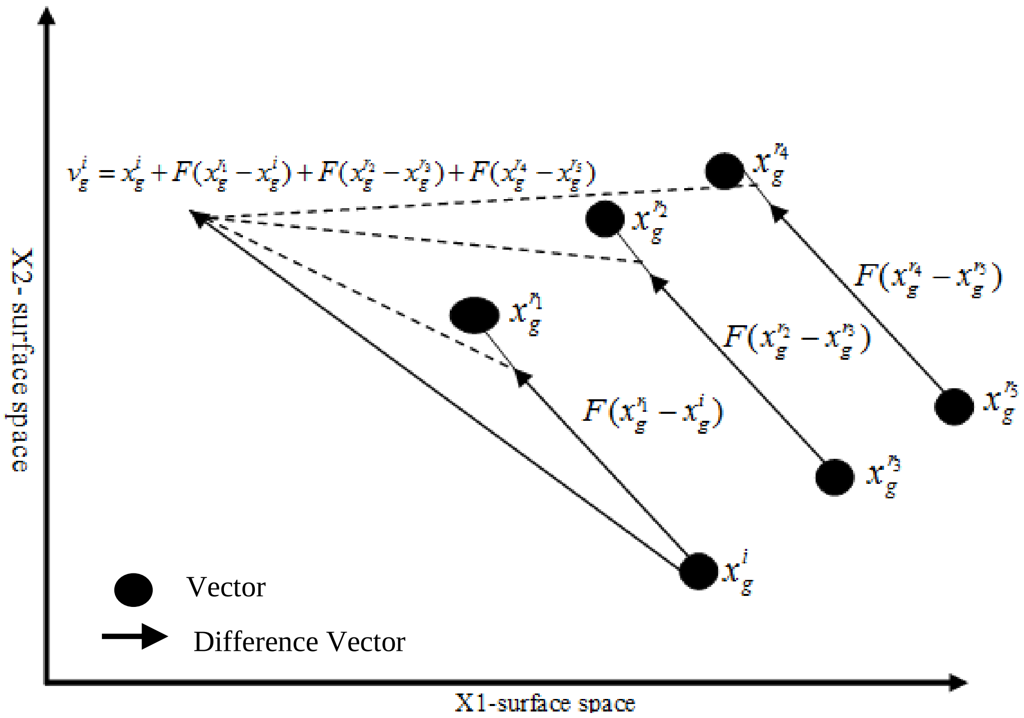


Figure.3.16 *DE/current to rand/2*

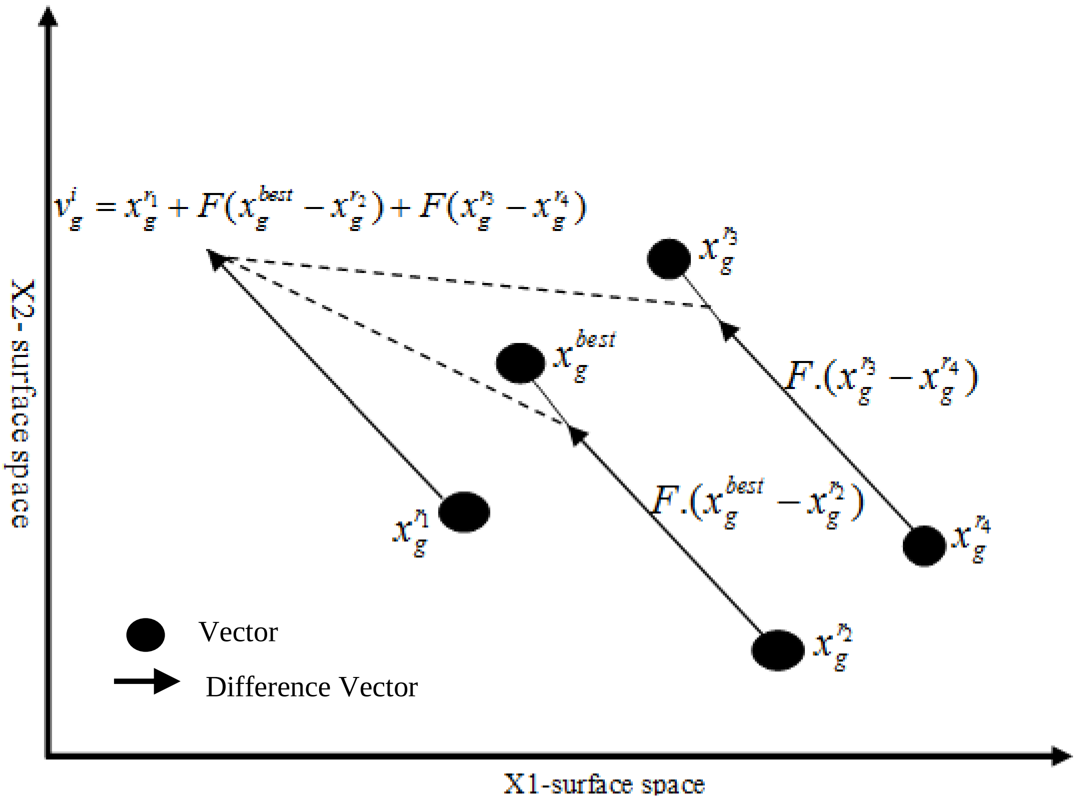


Figure.3.17 *DE/rand to best/2*

- ***"DE/rand to best/2"***

Podoba et. al **[**78**]** have used *"DE/rand to best/2"* mutation strategy in their research work on surface construction using AI. This mutation strategy has naming conflict with other mutation strategies in the literature (Discussed in chapter 4). The equation of *"DE/rand to best/2"* is

$$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4}) \quad \text{.......................................(3.17)}$$

"DE/rand to best/2" contains four random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4}$ and a best vector x_g^{best} in moving from random vector to a best vector and a difference vector of two random numbers. The graphical representation of this mutation strategy is given in figure 3.17.

- ***"DE/rand to best/2"***

Oliveira & Saramago **[**108**]** have used *"DE/rand to best/2"* mutation strategy in their research work. This mutation strategy has naming and equation conflicts with the other mutation strategies in the literature (Discussed in chapter 4). The equation of *"DE/rand to best/2"* is

$$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2}) \quad \text{.......................................(3.18)}$$

"DE/rand to best/2" contains two random vectors x_g^{r1}, x_g^{r2}, a best vector x_g^{best} and a current vector x_g^i in moving from current vector to best vector and a difference vector of two random vector. The graphical representation of this mutation strategy is given in figure 3.18.

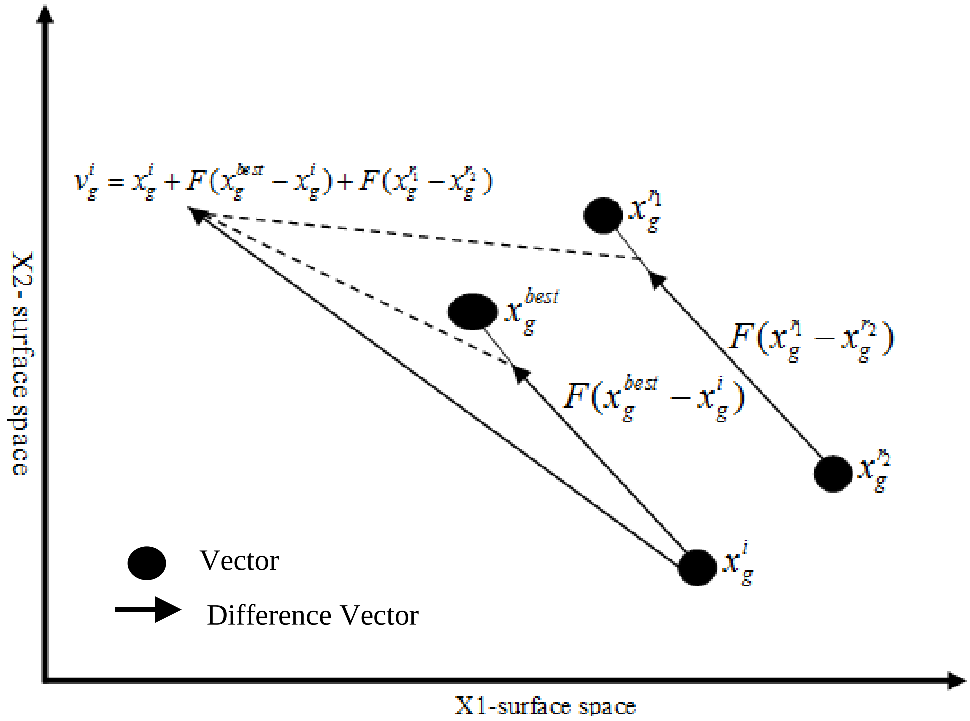


Figure.3.18 *DE/rand to best/2*

- ***"DE/rand to best/2"***

Weber **[**77**]** has used *"DE/rand to best/2"* mutation strategy in his PhD thesis on parallel global optimization. This mutation strategy has naming conflict with other mutation strategies in the literature (Discussed in chapter 4). The equation of *"DE/rand to best/2"* is

$$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^i) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5}) \quad \text{................(3.19)}$$

"DE/rand to best/2" contains five random vectors, $x_g^{r2}, x_g^{r3}, x_g^{r4}, x_g^{r5}$, a current vector x_g^i and a best vector x_g^{best} by adding a weighted moving from current vector to best vector in the random vector(x_g^{r1}) and two difference vector of random numbers. The graphical representation of this mutation strategy is given in figure 3.19.

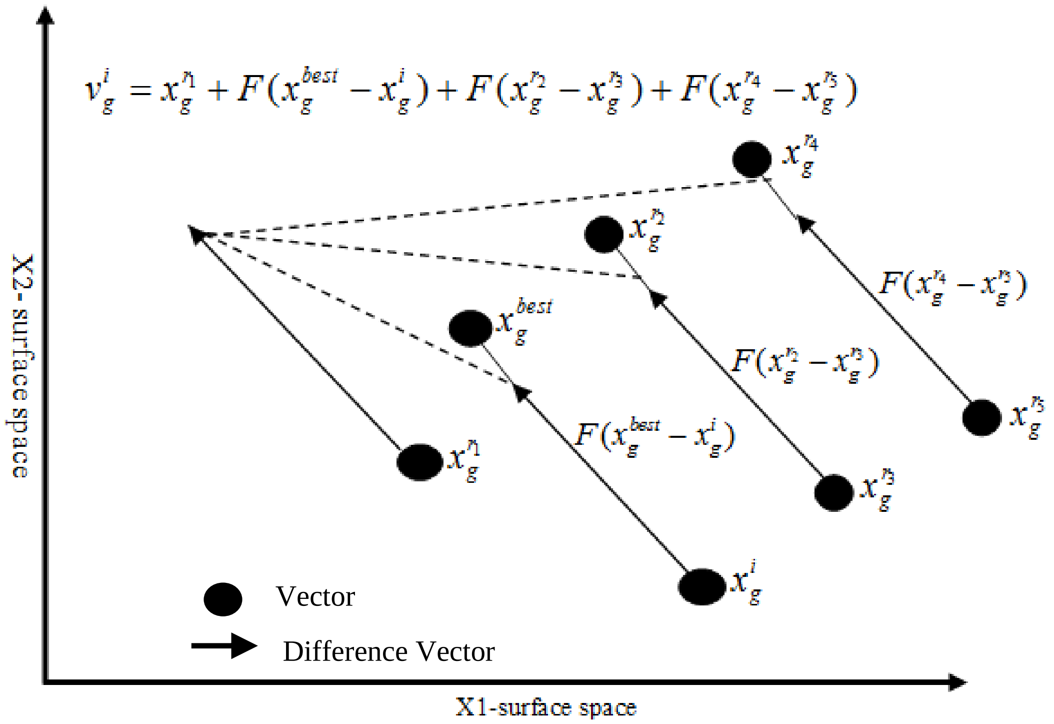


Figure.3.19 *DE/rand to best/2*

- ***"DE/rand to best/2"***

Triguero et. al **[**83**]**, Triguero et. al [84] and Goudos et. al [109] have used "*DE/rand to best/2"* mutation strategy in their research work. This mutation strategy has naming conflict with other mutation strategies in the literature (Discussed in chapter 4). The equation of "*DE/rand to best/2"* is

$$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4}) \quad \text{....................(3.20)}$$

"*DE/rand to best/2*" contains four random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4}$, a current vector x_g^i and a best vector x_g^{best} in moving from current vector to best vector and two difference vector of random vectors. The graphical representation of this mutation strategy is given in figure 3.20.

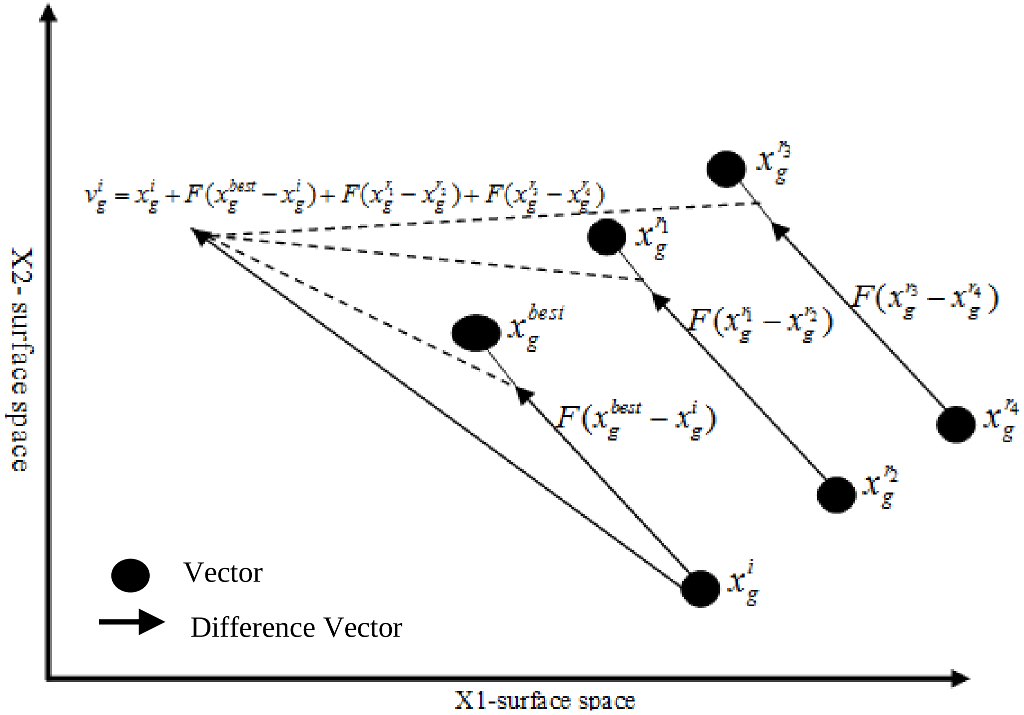


Figure.3.20 *DE/rand to best/2*

- ***"DE/rand to best/2"***

Gong & Cai [55], Gong et. al [57], Fialho et. al [110], Fialho et. al [111] have used *"DE/rand to best/2"* mutation strategy in their work. This mutation strategy has naming conflict with other mutation strategies in the literature (Discussed in chapter 4). The equation of *"DE/rand to best/2"* is

$$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r1}) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5}) \quad \text{.................(3.21)}$$

"DE/rand to best/2" contains five random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4}, x_g^{r5}$ and a best vector x_g^{best} in moving from a random vector to a best vector and two difference vector of random vectors. The graphical representation of this mutation strategy is given in figure 3.21.

- ***"DE/rand to current /2"***

Elsayed et. al [112] and Elsayed et. al [113] have used *"DE/rand to current /2"* mutation strategy in their research work. This mutation strategy has no conflict with the other mutation strategies in the literature (Discussed in chapter 4). The equation of *“DE/rand to current /2”* is

$$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^i) + F(x_g^{r3} - x_g^{r4}) \quad \text{.......................................(3.22)}$$

“DE/rand to current /2” contains four random vectors x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4} and a current vector x_g^i in moving from a current vector to random vector and a difference vector of two vectors. The graphical representation of this mutation strategy is given in figure 3.22.

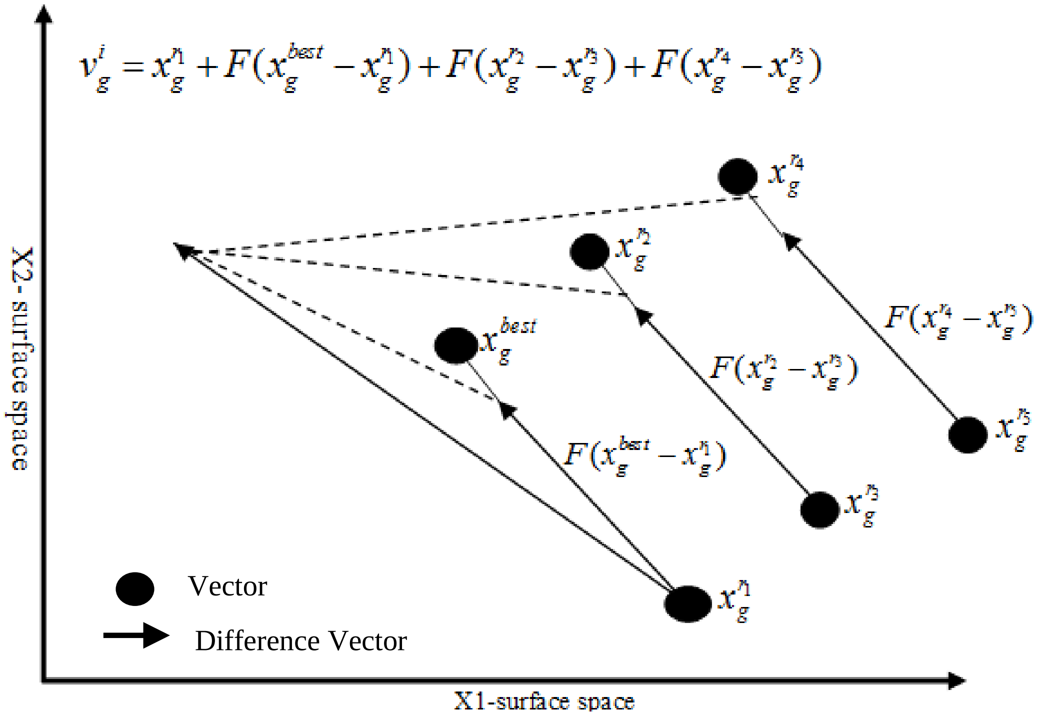


Figure.3.21 *DE/rand to best/2*

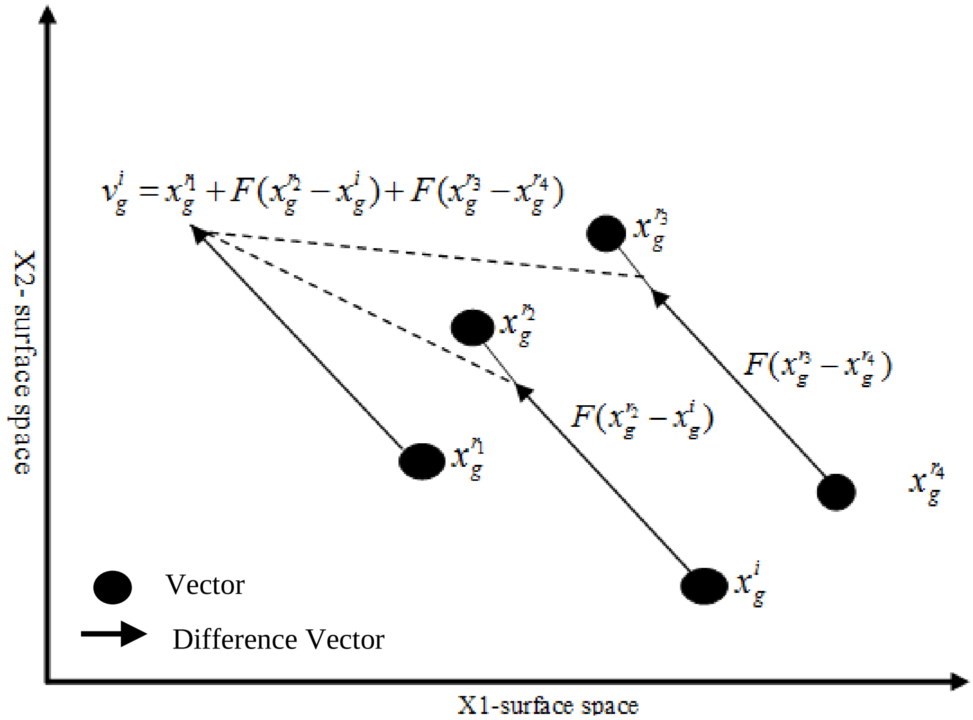


Figure.3.22 *DE/rand to current /2*

- ***"DE/rand to best and current /2"***

Elsayed et. al [112] and Elsayed et. al [113] have used *"DE/rand to best and current /2"* in their research work. This mutation strategy has no conflict with the other mutation strategies in the literature. The equation of *"DE/rand to best and current /2" is*

$$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r2}) + F(x_g^{r3} - x_g^i) \quad \text{……………………………(3.23)}$$

"DE/rand to best and current /2" contains three random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}$, a current vector x_g^i and a best vector x_g^{best} in moving from random vector to best vector and a difference vectors of random and current vectors. The graphical representation of this mutation strategy is given in figure 3.23.

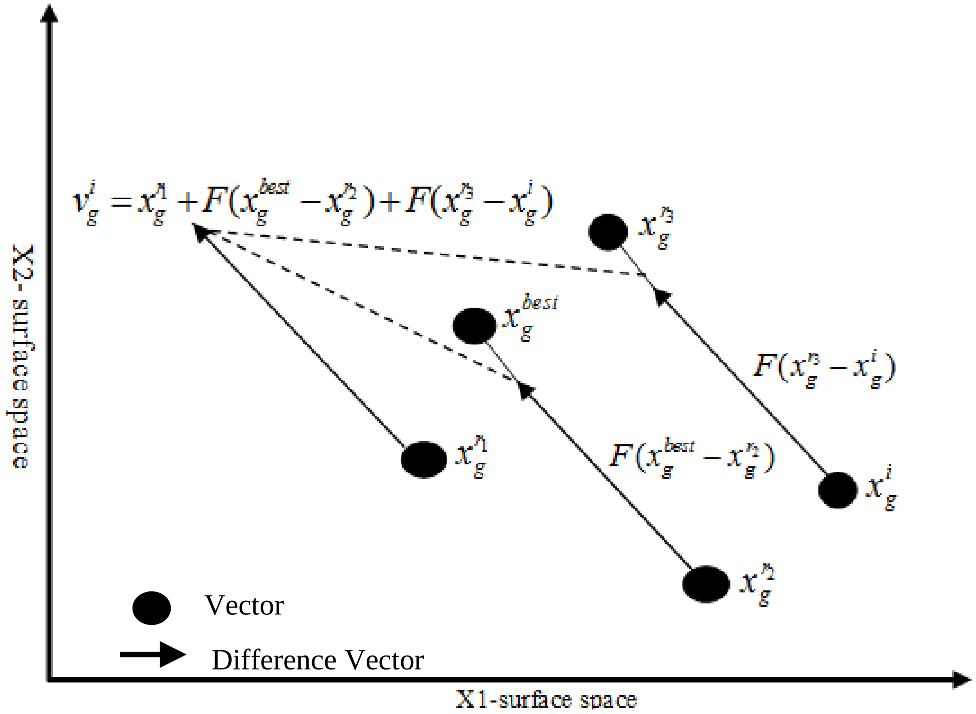


Figure.3.23 *DE/rand to best and current /2*

- ***"DE/mid-to-better/1"***

Xin et. al [50] have used *"DE/mid-to-better/1"* in their research work of designing powerful optimizer. This mutation strategy has no conflict with the other mutation strategies in the literature. The equation of *"DE/mid-to-better/1"* is

$$v_g^i = F(x_g^{better} + x_g^i)/2 + F(x_g^{better} - x_g^i) + F(x_g^{r1} - x_g^{r2}) \quad \text{……………………(3.24)}$$

"DE/mid-to-better/1" contains two random vectors x_g^{r1}, x_g^{r2}, a current vector x_g^i and a better vector x_g^{better}, one average vector by using current vector and a better vector and a moving

vector from current vector to a better vector and one difference vector of two random vectors. This variant is different from other variants in the sense that it is the only variant that uses better vector and average of better vector and current vector. The graphical representation of this mutation strategy is given in figure 3.24.

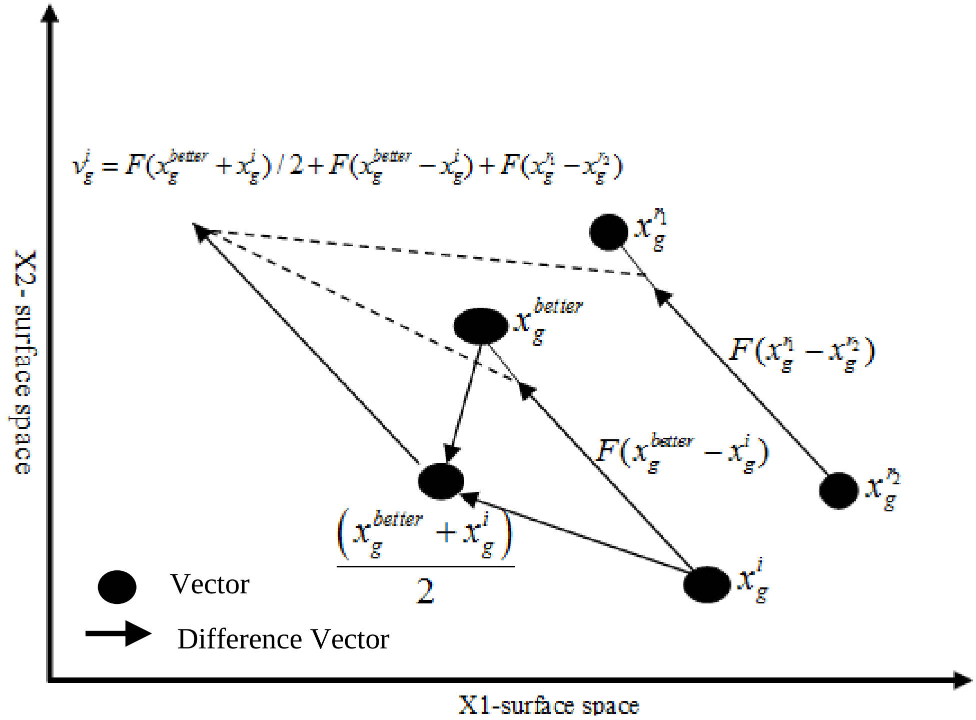


Figure.3.24 *DE/mid-to-better/1*

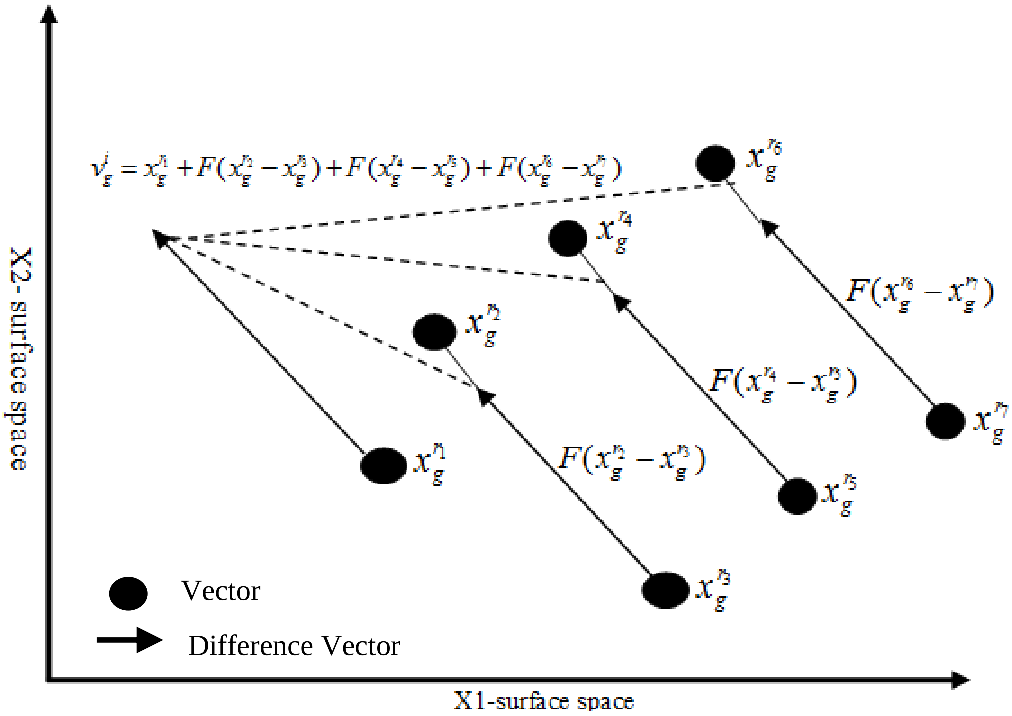


Figure.3.25 *DE/rand/3*

- ***“DE/rand/3”***

Elsayed et. al [112] and Elsayed et. al [113] have used “*DE/rand/3”* in their research work. This mutation strategy has no conflict with the other mutation strategies in the literature. The equation of “*DE/rand/3”* is

$$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5}) + F(x_g^{r6} - x_g^{r7}) \quad \text{..............................(3.25)}$$

“*DE/rand/3”* contains seven random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4}, x_g^{r5}, x_g^{r6}$ and x_g^{r7} in three difference vectors. The graphical representation of this mutation strategy is given in figure 3.25.

- ***“DE/best/3”***

Elsayed et. al **[**112**]** and Elsayed et. al [113] have used “*DE/best/3”* in their research work. This mutation strategy has no conflict with the other mutation strategies in the literature. The equation of “*DE/best/3” is*

$$v_g^i = x_g^{best} + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4}) + F(x_g^{r5} - x_g^{r6}) \quad \text{..........................(3.26)}$$

“*DE/best/3”* contains six random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4}, x_g^{r5}, x_g^{r6}$ and a best vector x_g^{best} in three difference vectors. The graphical representation of this mutation strategy is given in figure 3.26.

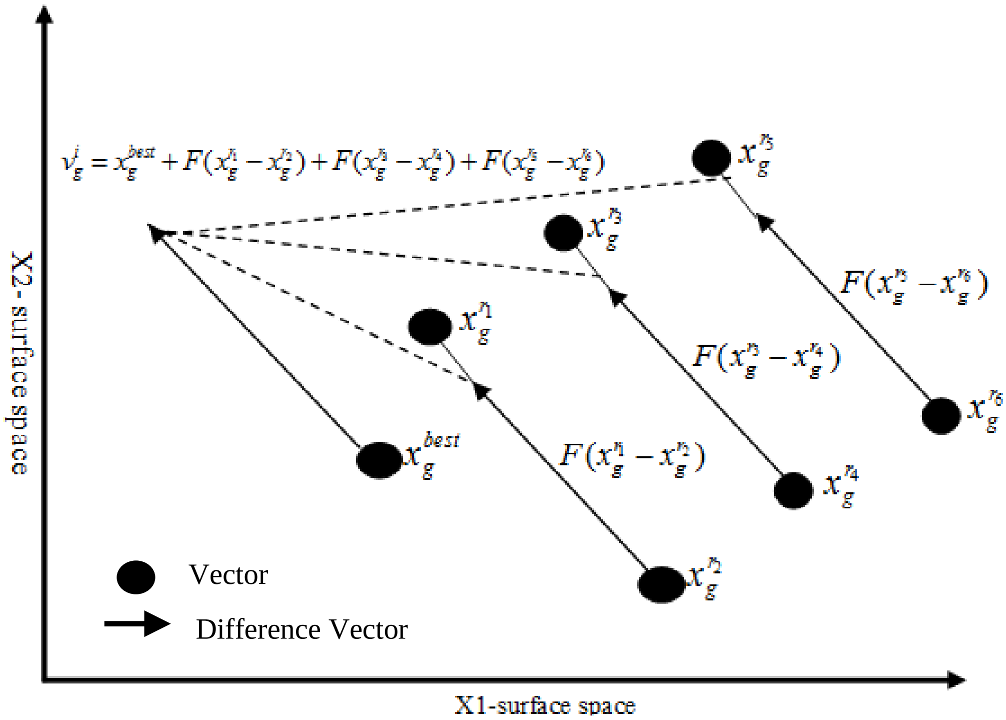


Figure.3.26 *DE/best/3*

3.2 Latest development in DE algorithm

Various researchers have made advancements in DE algorithm in terms of parameters and mutation strategies. This section presents recent developments in DE algorithm. DE algorithm is also used as a hybrid with other algorithms; the detail is given in this section.

3.2.1 Parameter and Mutation Strategies

This section focuses on the parameters and mutation strategies based advancements in DE algorithm. State of the art DE variants are based on the conventional DE variants. Parameter selection/adaption and strategy selection/adaption are the core part of the state of the art DE variants. Some researchers have used strategy pool by considering various DE mutation and crossover strategies in some version of state of the art DE variants. The strategy adaption state of the art DE variants prove themselves to be valuable in literature. The parameter and strategy adaption and self adaption are considered by many researchers in their research work. Zaharie [114] 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 [115] 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. To determine the control parameter F and CR values a new self adaptive version of DE algorithm (jDE) is introduced by Brest et al. [116]. This self adaptive DE proves to have promising results when compared with fuzzy adaptive DE (FADE) and other algorithms in their research work. The control parameter F and CR values are controlled by using two probabilities τ_1=0.1, τ_2=0.1respectively. A new parameter adaption method in introduced as an adaptive DE with optional archive by Zhang and Sanderson **[**117**]**. Their adaptive mechanism is based on *"DE/current to pbest/1"* strategy. They have used normal distribution and Cauchy distribution to generate the values of the control parameter CR and F respectively. To enhance the searching ability in DE algorithm, random best individual is selected from top 100% best population. *"DE/current to pbest/1"* is similar to conventional *"DE/current to best/1"* except that *pbest* is selected from top 100% best population instead of best of best vector. An optional archive is used that contain an inferior solution and then by using this concept new population is generated from the union of current population and archive. Control parameter self adaption and strategy selection self adaptive version of version of DE algorithm are proposed by Qin et al. **[**33**]** in their work. They have used self adaption of CR control parameter by using the concept of some previous generations learning period (LP). A strategy pool is created that is based on the four commonly used conventional strategies of DE algorithm with names *"DE/rand/1/bin", "DE/rand/2/bin", "DE/current-to-rand/1" and "DE/rand-to-best/2/bin"*. Mallipeddi et al. [35] have used pool of control parameter values and a pool of various mutation and crossover schemes in their research work. The pool of control parameter values uses predefined set of values for F and CR control parameters. The pool of strategies contains binomial and an exponential version of conventional strategy "DE/current-to-rand/1/bin" and JADE mutation strategy. In EPSDE parameter values and one mutation and crossover strategy are assigned to each vector to generate its corresponding target vector. In global local version of DE algorithm is introduced by Das et al. [118] that employs *"DE/target-to-best/1/bin"* conventional scheme. To balance the exploitation and exploration in DE searching they utilized the neighbourhood concept for each population member. In their research work donor vector is generated by combining local and global neighbourhood by using α and β are a scaling factor respectively. In local neighbourhood two random vectors are generated and best from their neighbourhood is selected while in global neighbourhood a best of best vector from the entire population is selected. Further a weight factor w and its variations are used to control exploitation and exploration in DE. A new mutation strategy *"DE/current-to-gr_best/1"* that is based on the convention variant *"DE/current-to-best/1"* is introduced by MinhazulIslam et al. [34]. New mutation strategy utilizes the concept of q% best population and select a best member from this population and named it *gr_best(group best)*. In their research work a conventional crossover strategy is modified as *pbest crossover*. In *pbest crossover* ptop-ranked individual's components can be exchanged with the mutant vector. They also have used control parameter adaption to control the values of F and CR using statistical distributions. Strategy adaptation mechanism(SaM) mechanism in introduced by Gong et al. [119] in their research work. In their research they have used the ensemble of JADE and SaM (SaJADE) with a strategy pool based on the conventional mutation and crossover strategies and JADE archive. The strategies they have used are *"DE/rand-to-pbest" with archive, "DE/current-to-pbest" without archive , "DE/current-to-pbest" with archive, "DE/rand-to-pbest" without archive* to form a strategy pool in their research work. A pool of three convention mutation and crossover strategies is used in composite DE (CoDE) by Wang, Cai and Zhand [120]. A pool of combination of various control parameter settings is also used in their research work with *[F= 1.0,Cr= 0.1]*, *[F= 1.0,Cr= 0.9]* and *[F= 0.8,Cr= 0.2]* combination. The conventional strategies used by these researches are *“rand/1/bin”, “current-to-rand/1” and “rand/2/bin”* that forms a strategy pool. Trial vector in CoDE is generated based on the selected strategy from the strategy pool and parameter values selected from the values pool.

Self adaptive learning based modification of DE algorithm in introduced by Li and Yin [121]. They have used two probabilistic rules to balance the exploration and exploitation in DE algorithm. The two rules are based on the random and best individuals of the population. They have used solution quality control parameter to assess the performance of self adaptive modified DE algorithm and compare it with the other state of the art algorithms. Guo et.al have incorporated the concept of successful-parent-selection framework in DE algorithm [122]. They have used achieve of successful solutions and then select parents from that achieve after an unacceptable amount of time when a solution is continuously not updated. They have compared their proposed method with well known state of the art algorithm as well as conventional DE mutation strategies. The ensemble of backtracking search optimization algorithm is incorporated in DE algorithm (E-BSADE) for function optimization application by Nama, Saha and Ghosh [123]. They have compared the average fitness value, successful performance and successful performance of E-BSADE with DE, BSA and other commonly used conventional strategies to show the significance of E-BSADE. Brown et. al **[**124**]** have introduced small population based concept in adaptive differential evolution. They have a population size less than 10 along with a new mutation operator that uses current, random and pbest vector. They have compared their proposed model with state-of-the-art algorithms that shows comparable performance with conventionally sized populations. Adaptation of the mutation scale factor concept in DE algorithm by Segura et. al [125]. The adaptation of the mutation scale factor focuses on the feedback from the search process. They have discussed the effectiveness of various adaptive schemes for exploration and exploitation in perspective of trial vector generation strategies. Gou et. al have proposed the concept of directional mutation strategy in DE algorithm [126**]**. They have used current fitness information to dynamically set the values of control parameters and trial vector generation pool. They have divided the whole population using distributed topology into three subgroups by assigning different crossover scheme and mutation strategies to each subgroup. The research result shows significant performance of the proposed algorithm. Parallel adaption based concept is implemented in DE algorithm for Systems Biology by Kozlov, Samsonov and Samsonova [127]. The deviation of solution from experimental data is minimized by identifying unknown parameters of mathematical models of systems biology. Dong et al. have incorporated the enhancement in local search of DE algorithm **[**128**]**. The enhancement combines the opposition-based learning and orthogonal crossover strategies. The enhanced local search based DE algorithm is applied to high dimensional optimization problem and shows significant performance. The concept of evolution path in DE algorithm (DEEP) is introduced by Li et al **[**129**]**. In DEEP the cumulatively learned evolution path and covariance matrix adaptation concept is used to evolve the current population. The function optimization application is used to assess the performance of DEEP algorithm. Individual-dependent mechanism in DE algorithm is implemented by Tang, Dong and Liu **[**130**]**. The individual-dependent mechanism combined individual-dependent mutation strategy and individual-dependent parameter for function optimization application. The individual-dependent parameter are assigned to population individual while individual-dependent mutation strategy contains 4 well know mutation strategies having distinct characteristics. Rank based scheme and value based scheme are used for F and CR control parameters. Gong, Cai and Liang **[**131**]** have implemented adaptive ranking mutation operator in DE algorithm. The proposed algorithm is applied on three situations the infeasible situation, semi-feasible situation and feasible situation based population. The ranking of solutions in the infeasible, semi feasible and feasible are ranked based on the constraint violations, transformed fitness and objective function value respectively. Biswas, Kundu and Das have introduced niche based concept is used in DE algorithm **[**132**]**. The elitism based concept in many-objective optimization DE algorithm is proposed by Bandyopadhyay and Mukherjee **[**133**]**. The elitism based concept focuses on the restriction of higher ranked individuals in the next generation. Gao et al. have introduced convolution based dual population concept of in DE algorithm (DPDE) for constrained optimization problem **[**134**]**. The search information between the subpopulations is exchanged by using information sharing strategy in DPDE algorithm. DPDE considers two objectives naming as actual cost and degree of constraint violations. Both these objectives are treated as different during the evolutionary process at each generation. Elsayed, Sarker and Essam have introduced the improved DE algorithm that uses the concept of multi operator **[**135**]**. Their proposed algorithm divided the main population into a number of subpopulations. They have assigned each sub population to each operator of DE algorithm. The two crossover operators and one mutation strategy were used with each sub population during the evolutionary process.

3.2.2 DE as a hybrid Algorithm

Many researchers have hybridized DE algorithm with other algorithms. Lu et.al have hybridized DE algorithm with silhouette filter in maximizing the accuracy of tumor pattern classification and search the optimal gene subset of microarray data **[**136**]**. Variable Rate Particle Filter (VRPF) model is hybridized by Saeidi, Moniri and Shahzadi with DE algorithm to solve the problem of trajectory control **[**137**]**. VRPF faces the problem of tracking an object in the maneuvering regions that is minimized using DE algorithm by obtaining high performance results as compared to VRPF. Gao, Zhu and Lang **[**138**]** have introduced improved DE algorithm by incorporating the concept of hybrid multi-strategy (HMSDE) in mutation strategies. In order to dynamically control crossover factor and scaling factor the dynamic adaptive strategy is used; the concept of multi population is used to maintain diversity in the population; the information exchange among sub populations is achieved by synergy strategy. De et al. have hybridized DE algorithm with the Harmony Search for optimal design of high speed symmetric switching CMOS inverter **[**139**]**. Li and Yin have hybridized DE algorithm with an artificial bee colony algorithm for parameter estimation of chaotic systems [140]. They have combined the exploration ability of DE algorithm and the exploitation ability of artificial bee colony algorithm in their research work. The result of average fitness performance parameter shows the significant performance of proposed hybridization for parameter estimation of chaotic systems. Yildiz has presented the hybrid of Taguchi's method with DE algorithm in his research work to minimize the production cost of multi-pass turning problem **[**141**]**. He has also refined the current population instead of refining the search space in his research. Research result shows the significance of this hybridization. Fister et al. have used hybrid of ANN with DE algorithm for the prediction of successful mutation strategy of DE algorithm **[**142**]**. In this hybridization ANN is used to predict the best performing mutation strategy using regression in the current generation that is to be used in the next generation. Experimental results shows the significance of such a prediction based hybridization. Bazi et al. have introduced DE based extreme learning machine (ELM) for the classification problem. Early on ELM was introduced in the training of a single hidden-layer feed forward neural networks **[**143**]**. Their research work hybridized ELM and DE for byperspectral images classification that shows the significance results. Wang et al. have hybridized DE algorithm with krill herd algorithm in their research work for function optimization application **[**144**]**. The DE algorithm helps krill herd algorithm to perform local search with the defined search region. Their proposed hybridization improves the convergence speed and accuracy of results. Yildiz has presented the hybrid of artificial immune system algorithm with DE algorithm in his research work [145]. He has used receptor editing property artificial immune system algorithm along with DE algorithm in the milling operations to find the optimal machining parameters. Research result shows the significant performance when compared with other well known evolutionary algorithms.

3.3 DE current applications

DE algorithm is one the widely used evolutionary computing algorithm. Zamee et al. [146] have applied DE algorithm on Hydro-Thermal power plants to achieve optimal value of gain parameters. They have considered renewable energy and two-area conventional aspects of this nonlinear power system. DE algorithm is helpful to improve the performance of hydrothermal power by getting better optimal values of gain parameters. To solve the low cross polarization synthesis problem of conformal array, Li et. al have developed a Competition Differential Evolution Strategy (CDES) algorithm [147]. The research result shows that CDES has significant performance in solving polarization synthesis problem. The concept of mixed strategies DE algorithm [148] to control the open quantum system in their research work is used by Ma, Chen and Chen. Against parameter fluctuations in the quantum system, they have used sampling-based learning control approach by generating artificial sample to design an optimal control field. Sumithra and Victoire have applied DE algorithm for Wireless Sensor Networks. The vicinity based operator is added in DE in Optimal Routing and Clustering of Energy Efficient [149]. The tradeoff between energy consumptions of the cluster heads using diversified vicinity procedure in DE algorithm. Kundu et. al [150] have applied improved DE algorithm for lifetime maximization of wireless sensor networks. They focused on the combined life time maximization of sleep scheduling scheme and combined routing are considered in their research work. They have also proposed modified semi-adaptive DE variant (MSeDE). The hybrid of DE algorithm with common sub expression elimination algorithms for hardware efficient finite impulse response application is introduced by Reddy and Sahoo [151]. By hybridizing DE algorithm the number of signed-power-of-two is reduced without losing the quality of the filter response. Mohanty, Panda and Hota [152] have applied controller parameters tuning base DE algorithm to a multi-source power for load frequency control. The multi-source power of thermal, hydro and gas power plants are considered in their research. Various mutation strategies are fine tuned during the evolutionary process considered in this research for the above discusses applications. Jiang has implemented the concept of multi-population and self-adaptive behaviour in DE algorithm [153]. In the first step of the algorithm the population is divided into multi-populations and then the self adaptive method is used to control the values of DE parameters in order to main the balance between local search and the global search.

3.4 Limitations in the current literature

For varying nature of problems, DE algorithm has some limitations that are discussed in the following sections

3.4.1 Local optima/premature convergence

DE algorithm faces the problem of premature convergence for the problems having varying nature that can be either because of biased mutation strategies, less exploration and diversity issue in the population [12,34,66,85,118,138,154-161]. This limitation can be minimized by keeping focus on more fitter as well as less fit individuals in DE mutation strategies that will be helpful to escape from local optima.

3.4.2 Population diversity and convergence speed

Population diversity is very important aspect of differential evolution algorithm **[**132**]**. Population diversity is helpful to increase the exploration ability of differential evolution algorithm **[**155**]**. Exploration ability helps differential evolution algorithm to escape from local optima and to reach at global optimum value or near optimal solution **[**34**]**. DE algorithm faces the problem of population diversity that can be because of biasness towards global optimum, formation of mutation strategy or inappropriate parameter values [12, 34, 66, 85, 118, 120, 154, 158]. Convergence speed is another challenging issue of differential evolution algorithm. The slow convergence in DE algorithm can be either due to the diverse nature of mutation strategy, formation of mutation strategy, inappropriate values of DE parameters like (F, CR, NP) or Nature of the problem(like problems having multiple peaks, non-separable etc)[34, 66, 85,120,161-162]. This limitation can be minimized by introducing powerful mutation strategy that will be helpful to maintain diversity in the population and improve the convergence speed of DE algorithm that will be helpful to escape from local optima and reach at global optimal value quickly.

3.4.3 DE mutation strategies inconsistencies:

Various mutation strategies in the literature restrain the irregularities with respect to naming and mathematical equation. DE algorithm variants should be misinterpretation free for its prosperity as an algorithm. To make this algorithm problem free an effort is made in this research to identify and remove the inconsistencies associated with DE algorithm mutation strategies in literature. Inconsistencies in DE mutation strategies are identified with respect to their names and equations. The detail of these inconsistencies is discussed in chapter 4 of this thesis.

Chapter # 4: Identification and Removal of DE mutation strategies inconsistencies

4.0 Chapter Summary

This chapter presents identification inconsistencies in DE mutation strategies. The inconsistencies are discussed in the term of mathematical formulation and the naming of mutation strategies. The solution to remove these inconsistencies is also discussed in this chapter.

4.1Identification of mathematical equation inconsistencies in DE mutation variants

Mathematical equation of variants has key role since the implementation of a variant is carried out according to its mathematical equations. Numerous variants in the literature contain same mathematical equations, but different names which create inconsistencies and lead to false impressions. In this section variants of table-1 having *"same mathematical equations"* and *"different names"* are identified. The detail is given as follows:

(*In section 4.1, 4.2 and 4.3, the mutation strategies are accessed by the serial index of table-3.1*)

- Mutation strategies 8 and 9 have same mathematical equation and same graph (Figure 3.8, 3.9) but different names "*DE/current to best/1*" & "*DE/rand to best/1*" respectively
- Mutation strategies 7, 10, 14 and 18 have same mathematical equations and same graph (Figure 3.7, 3.10, 3.14, 3.18) but different names *"DE/current to best/1", "DE/rand to best/1", "DE/current to best/2" and "DE/rand to best/2"* respectively
- Mutation strategies 11and 17 have same mathematical equations and same graph (Figure 3.11, 3.17) but different names *"DE/rand to best/1" and "DE/rand to best/2"*
- Mutation strategies 15 and 20 have same mathematical equations and same graph (Figure 3.15, 3.20) but different names "DE/current to best/2" and "DE/rand to best/2"

4.2 Identification of naming inconsistencies in DE mutation variants

There are many variants in the literature having same name but different mathematical equations that fabricate misunderstanding for the researchers. In this section mutation strategies of table-3.1 having *"same names"* and *"different mathematical equations"* are identified. The detail of the naming inconsistencies is as follows

- Mutation strategy 5 and 6 have the same name *"DE/current to rand/1"* but different mathematical equations and different trial vector generation graphs (Figure 3.5 -3.6).

- The same name *"DE/current to best/1"* is used for the mutation strategies 7 and 8 but different equations and different trial vector generation graphs (Figure 3.7 - 3.8).
- *"DE/rand to best/1"* name is used for variants 9, 10, 11, 12 and 13 but these mutation strategies have different equations and different trial vector generation graphs (Figure 3.9-3.13).
- The two mutation strategies 14 and 15 share the same name *"DE/current to best/2"* but have different equations and different trial vector generation graphs (Figure 3.14-3.15).
- Four different mutation strategies 17, 18, 19, 20 and 21 have the same name *"DE/rand to best/2"* but different equations and different trial vector generation graphs (Figure 3.17-3.21).

Table 4.1: List of DE mutation strategies/variants available in the Literature

S. No	**Binomial/ Exponential**		**S. No**	**Binomial/ Exponential**	
	Variant Name	**Equation**		**Variant Name**	**Equation**
V_1	DE/rand/1	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3})$	V_{11}	DE/rand to best/1	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^i) + F(x_g^{r2} - x_g^{r3})$
V_2	DE/best/1	$v_g^i = x_g^{best} + F(x_g^{r1} - x_g^{r2})$	V_{12}	DE/current to best/2 & DE/rand to best/2	$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4})$
V_3	DE/rand/2	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$	V_{13}	DE/current to rand/2	$v_g^i = x_g^i + F(x_g^{r1} - x_g^i) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$
V_4	DE/best/2	$v_g^i = x_g^{best} + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4})$	V_{14}	DE/rand to best/2	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^i) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$
V_5	DE/current to rand/1	$v_g^i = x_g^i + F(x_g^{r1} - x_g^i) + F(x_g^{r2} - x_g^{r3})$	V_{15}	DE/rand to best/2	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r1}) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$
V_6	DE/Current-to-rand/1	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^i) + F(x_g^{r1} - x_g^{r3})$	V_{16}	DE/rand to current /2	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^i) + F(x_g^{r3} - x_g^{r4})$
V_7	DE/current to best/1 & DE/rand to best/1 & DE/current to best/2 & DE/rand to best/2	$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2})$	V_{17}	DE/rand to best and current /2	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r2}) + F(x_g^{r3} - x_g^i)$
V_8	DE/current to best/1 & DE/rand to best/1	$v_g^i = x_g^i + F(x_g^{best} - x_g^{r1}) + F(x_g^{r1} - x_g^{r2})$	V_{18}	DE/mid-to-better/1	$v_g^i = F(x_g^{better} + x_g^i)/2 + F(x_g^{better} - x_g^i) + F(x_g^{r1} - x_g^{r2})$
V_9	DE/rand to best/1 & DE/rand to best/2	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4})$	V_{19}	DE/rand/3	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5}) + F(x_g^{r6} - x_g^{r7})$
V_{10}	DE/rand to best/1	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r1}) + F(x_g^{r2} - x_g^{r3})$	V_{20}	DE/best/3	$v_g^i = x_g^{best} + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4}) + F(x_g^{r5} - x_g^{r6})$

4.3 The proposed Scheme to remove Inconsistencies

The mutation strategies having same mathematical equations, but different names are combined because mutation strategies having same equation produced same representation and same results that create problems for the user due to their different names. Mutation strategies having same equations and different names are combined and reported in table-4.1, after combing these mutation strategies they have assigned unique names refereeing the equation of that mutation strategy and reported in table-4.2. Mutation strategies 8 and 9 are combined as V_8 because these mutation strategies have same mathematical equation in table-4.1; variants 7, 10, 14 and 18 are combined in table-4.1 as V_7; mutation strategies 11 and 17 are merged as V_9 in table-4.1; mutation strategies 15 and 20 are mingled as V_{12} in table-4.1.

The short names of DE mutation strategies/variants that will be used throughout the remaining thesis where V_1 to V_{20} represents the binomial and V_{21} to V_{40} exponential. Mutation strategies V_{21} to V_{40} are exponential mutation strategies of corresponding V_1 to V_{20} binomial mutation strategies respectively.

Mathematical equation inconsistencies in DE mutation strategies are removed from table-3.1 (as discussed in section **4.2**) and reported in table-4.1. Table-4.1 still contains naming inconsistency like name *"DE/rand-to-best/1"* is used by more than one mutation strategy; Naming inconsistencies are removed from table-4.1 and new names are suggested to the ambiguous mutation strategies having similar name but different equations, resultant mutation strategies are reported in table-4.2. Later on the mutation strategies of table-4.2 are used in the thesis to generate the results.

The naming inconsistency is removed by suggesting the consistent names in table-4.2. The suggested names are relative to the equation of variant by considering the number of vectors, base vector and arrangement of vectors used to form the equation of the variant.

1) The variant V_5 has the same name as of variant V_6.
   - V_5 with name "DE/current to rand/1" have the following equation

$$v_g^i = x_g^i + F(x_g^{r1} - x_g^i) + F(x_g^{r2} - x_g^{r3}) \quad \text{..(4.1)}$$

The equation of this variant places the perturbation at a location between the current population member x_g^i and a randomly chose population members . One weighted difference vector is used in the equation of the variant. The equation of V_5 confirms the name of the mutation strategy/variant "*DE/current to rand/1"*.

   - The name "*DE/current to rand/1"* is used for mutation strategy V_6 with the following equation $v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^i) + F(x_g^{r1} - x_g^{r3})$ ....(4.2)

     "*DE/current to rand/1"* places the perturbation at a location away from a random vector x_g^{r1} between current vector and other random vector. This scheme contains three random vectors x_g^{r1} , x_g^{r2} , x_g^{r3} and a current vector x_g^i in the perturbation by repeating a random number x_g^{r1} in the weighted difference vector. The suitable name of this variant relative to its equation is *"DE/rand repeating & current to rand /1"* because a base random vector x_g^{r1}

repeats in the weighted difference vector and the perturbation location is towards location between current vector and a random vector.

- The mutation strategies V_7, V_8, V_9, V_{10} and V_{11} have naming inconsistency with each other.
    - Mutation strategy V_7 is used in the literature with various names: "*DE/current to best/1*", *"DE/rand to best/1"*, *"DE/current to best/2"* & *"DE/rand to best/2"*. The equation of V_7 is
    $$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2}) \quad \text{.....(4.3)}$$
    The equation of this mutation strategy places the perturbation at a location between the current vector x_g^i and best vector at current generation x_g^{best}. The mutation strategy uses one weighted difference of random vectors x_g^{r1}, x_g^{r2} in perturbing the current vector x_g^i so the suitable name for this mutation strategy relative to its equation is "*DE/current to best/1*".
    - The names *"DE/current to best/1" & "DE/rand to best/1"* are used for mutation strategies V_8 with the following equation
    $$v_g^i = x_g^i + F(x_g^{best} - x_g^{r1}) + F(x_g^{r1} - x_g^{r2}) \quad \text{.....(4.4)}$$
    The equation of this mutation strategy perturbs current vector x_g^i by moving perturbation at a location between best vector x_g^{best} and a random vector x_g^{r1}. The mutation strategy utilizes two random vectors x_g^{r1}, x_g^{r2}, a best vector x_g^{best} and a current vector x_g^i in perturbation with one weighted difference vector by repeating a random vector x_g^{r1} so the suitable name of this mutation strategy relative to its equation is *"DE/current & rand repeating to best/1"*.
    - *"DE/rand to best/1" & "DE/rand to best/2"* names are used in the literature for mutation strategy V_9 with equation
    $$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4}) \quad \text{.....(4.5)}$$
    The equation of this mutation strategy perturbs random vector x_g^{r1} by moving the perturbation at location between best vector x_g^{best} and random vector x_g^{r2}. This perturbation uses four random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}$, x_g^{r4} and a best vector x_g^{best} with one weighted difference vector so the suitable name of this mutation strategy relative to its equation is *"DE/rand to best/1"*.

- The mutation strategy V_{10} is used in the literature with name *"DE/rand to best/1"* that is same name as of mutation strategy V_9. The equation of V_{10} is

$$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r1}) + F(x_g^{r2} - x_g^{r3}) \quad \text{...(4.6)}$$

  The equation of this variant perturbs a random vector x_g^{r1} by moving the perturbation at location between x_g^{r1} and best vector x_g^{best}. This perturbation contains three random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}$ and a best vector x_g^{best} with a weighted difference vector. The suggested name of this mutation strategy relative to its equation is *"DE/rand repeated to best/1"*.

- Another mutation strategy V_{11} has name *"DE/rand to best/1"* in the literature. The same name *"DE/rand to best/1"* is also used by many other mutation strategies as discussed above in this section. The equation of V_{11} is

$$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^i) + F(x_g^{r2} - x_g^{r3}) \quad \text{...(4.7)}$$

  The equation of this mutation strategy perturbs random vector x_g^{r1} by moving the perturbation at location between best vector x_g^{best} and current vector x_g^i. This perturbation contains three random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}$, a current vector x_g^i and a best vector x_g^{best} with one weighted difference vector. The suggested name of this mutation strategy relative to its equation is *"DE/rand & current to best/1"*.

- The mutation strategies V_{12}, V_{14}, V_{15} have naming inconsistency with one another.
  - V_{12} are used in the literature with names *"DE/current to best/2"* & *"DE/rand to best/2"*. The equation of V12 is

$$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4}) \quad \text{.................................(4.8)}$$

    The equation of this mutation strategy perturbs current vector x_g^i by moving the perturbation at location between best vector x_g^{best} and current vector x_g^i. This perturbation contains four random vectors $x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4}$ a best vector x_g^{best} and a current vector x_g^i with two weighted difference vectors. The suggested name of this mutation strategy relative to its equation is *"DE/current to best/2"*.

  - The name *"DE/rand to best/2"* is used in the literature for mutation strategy V_{14}. The same name *"DE/rand to best/2"* is also used for other mutation strategies (like V_{12}, V_{15}) in the literature. The equation of V_{14} is

$$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^i) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5}) \quad \text{.................................(4.9)}$$

The equation of this mutation strategy perturbs random vector x_g^{r1} by moving its perturbation location towards location between best vector x_g^{best} and random vector x_g^i. This perturbation contains five random vectors, x_g^{r2}, x_g^{r3}, x_g^{r4}, x_g^{r5}, a current vector x_g^i and a best vector x_g^{best} with two weighted difference vectors. The suggested name of this mutation strategy relative to its equation is *"DE/rand & current to best/2"*.

- *"DE/rand to best/2"* name is used for mutation strategy V_{15} is used in the literature while the same name is used for some other mutation strategies as well (like V_{12}, V_{14}). The equation of V_{15} is

$$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r1}) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5}) \quad \text{(4.10)}$$

The equation of this mutation strategy perturbs random vector x_g^{r1} by moving the perturbation at a location between best vector x_g^{best} and same random vector x_g^{r1}. This perturbation contains five random vectors x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4}, x_g^{r5} and a best vector x_g^{best} with two weighted difference vectors. The suitable name of this mutation strategy relative to its equation is *"DE/rand repeated to best/2"*.

- The mutation strategy V_{16} is used in the literature with name *"DE/rand to current /2"*. The equation of V_{16} is

$$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^i) + F(x_g^{r3} - x_g^{r4}) \quad \text{(4.11)}$$

The equation of this mutation strategy perturbs random vector x_g^{r1} by moving its perturbation location towards the location between random vector x_g^{r2} and random vector x_g^i. This perturbation contains four random vectors x_g^{r1}, x_g^{r2}, x_g^{r3}, x_g^{r4} and a current vector x_g^i with one difference vector. The suggested name of this mutation strategy relative to its equation is *"DE/rand & current to rand /1"*.

- *"DE/rand to best and current /2"* name is used for the mutation strategy V_{17} in the literature. The equation of V_{17} is

$$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r2}) + F(x_g^{r3} - x_g^i) \quad \text{(4.12)}$$

The equation of this mutation strategy perturbs random vector x_g^{r1} by moving its perturbation location towards the location between best vector x_g^{best} and random vector x_g^{r2}. This perturbation contains three random vectors x_g^{r1}, x_g^{r2}, x_g^{r3}, a current vector x_g^i

and a best vector x_g^{best} with one difference vector that utilizes current vector x_g^i. The suggested name of this mutation strategy relative to its equation is *"DE/rand to best and current /1"*.

This chapter concludes that mutation strategies in the literature restrain the irregularities with respect to the naming of the mutation strategies and mathematical equation of mutation strategies. DE algorithm mutation strategies should be misinterpretation free for its prosperity as an algorithm; to make this algorithm problem free an effort is made in this research to identify and remove the inconsistencies associated with DE algorithm mutation strategies in the literature.

Table 4.2: List of DE mutation strategies/ variants after removal of naming inconsistencies

S. No	**Binomial/ Exponential**			S. No	**Binomial/ Exponential**		
	Strategy/Variant Current Name	Variant Proposed Name	Equation		Variant Current Name	Strategy/Variant Proposed Name	Equation
V_1	DE/rand/1	DE/rand/1	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3})$	V_{11}	DE/rand to best/1	DE/rand & current to best/1	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^i) + F(x_g^{r2} - x_g^{r3})$
V_2	DE/best/1	DE/best/1	$v_g^i = x_g^{best} + F(x_g^{r1} - x_g^{r2})$	V_{12}	DE/current to best/2 & DE/rand to best/2	DE/current to best/2	$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4})$
V_3	DE/rand/2	DE/rand/2	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$	V_{13}	DE/current to rand/2	DE/current to rand/2	$v_g^i = x_g^i + F(x_g^{r1} - x_g^i) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$
V_4	DE/best/2	DE/best/2	$v_g^i = x_g^{best} + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4})$	V_{14}	DE/rand to best/2	DE/rand & current to best/2	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^i) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$
V_5	DE/current to rand/1	DE/current to rand/1	$v_g^i = x_g^i + F(x_g^{r1} - x_g^i) + F(x_g^{r2} - x_g^{r3})$	V_{15}	DE/rand to best/2	DE/rand repeated to best/2	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r1}) + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5})$
V_6	DE/Current-to-rand/1	DE/rand repeating & current to rand /1	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^i) + F(x_g^{r1} - x_g^{r3})$	V_{16}	DE/rand to current /2	DE/rand & current to rand /1	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^i) + F(x_g^{r3} - x_g^{r4})$
V_7	DE/current to best/1 & DE/rand to best/1 & DE/current to best/2 & DE/rand to best/2	DE/current to best/1	$v_g^i = x_g^i + F(x_g^{best} - x_g^i) + F(x_g^{r1} - x_g^{r2})$	V_{17}	DE/rand to best and current /2	DE/rand to best and current /1	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r2}) + F(x_g^{r3} - x_g^i)$
V_8	DE/current to best/1 & DE/rand to best/1	DE/current & rand repeating to best/1	$v_g^i = x_g^i + F(x_g^{best} - x_g^{r1}) + F(x_g^{r1} - x_g^{r2})$	V_{18}	DE/mid-to-better/1	DE/mid-to-better/1	$v_g^i = F(x_g^{better} + x_g^i)/2 + F(x_g^{better} - x_g^i) + F(x_g^{r1} - x_g^{r2})$
V_9	DE/rand to best/1 & DE/rand to best/2	DE/rand to best/1	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4})$	V_{19}	DE/rand/3	DE/rand/3	$v_g^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3}) + F(x_g^{r4} - x_g^{r5}) + F(x_g^{r6} - x_g^{r7})$
V_{10}	DE/rand to best/1	DE/rand repeated to best/1	$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r1}) + F(x_g^{r2} - x_g^{r3})$	V_{20}	DE/best/3	DE/best/3	$v_g^i = x_g^{best} + F(x_g^{r1} - x_g^{r2}) + F(x_g^{r3} - x_g^{r4}) + F(x_g^{r5} - x_g^{r6})$

Chapter # 5: A Novel Tournament Selection based Enhancement to DE Algorithm

5.0 Chapter Summary

This chapter presents a novel tournament parent selection based mutation strategy of DE algorithm. The proposed mutation strategy is tested with standard benchmark functions in terms of average fitness value and number of function call performance parameters. The proposed mutation strategy is also compared with other well known strategies in DE algorithm as well as Particle swarm optimization and Genetic algorithm.

5.1 A Novel tournament selection base mutation strategy of DE algorithm

This thesis focuses on a novel mutation strategy of DE algorithm. This mutation strategy focuses on the tournament base parent selection of parent vectors used in the equation mutation. This tournament selection based mutation strategy of DE (TSDE) algorithm is more robust in terms of convergence speed and solution quality. The detail of proposed mutation strategy is given as follows

5.1.1 TSDE variant

Tournament selection is one of the famous selection approach used in Genetic Algorithms **[**163**]**. The parameter associated with this scheme is the size of the tournament that selects the number of individuals to participate in the competition of the tournament. To decrease the risk of premature convergence the loss of diversity should be kept as low as possible in the population [164]. To enhance the performance of population based algorithms, various researches have used tournament selection in their research work [165-66]. The DE algorithm sometimes faces the problem of slow and/or premature convergence [167]. Exploration and exploitation are very important aspects that are helpful in improving the convergence acceleration and solution quality of evolutionary algorithms [168]. The mutation strategies “DE/rand/1” and “DE/rand/2” are helpful for exploration [33] and *“DE/best/1”*, *“DE/rand-to-best”* and *“DE/best/2”* are more helpful for exploitation than exploration due to the less population diversity [169]. To balance the exploration and exploitation ability of proposed TSDE variant both random as well as best vectors are used. A novel DE mutation strategy based on the selection of parent is introduced in this research work. The new mutation strategy utilizes the knowledge of both best performing vector(s) and random vectors in creating the mutant vector. Like mutation strategies “DE/rand/1” and “DE/rand/2” the proposed mutation strategy uses random vector as a base vector along with two difference vectors. To generate a mutant vector in DE each difference vector utilizes the knowledge of one distinct best vector that will be helpful in incorporating diversity and improving convergence in the proposed mutation strategy. The two difference vectors and two distinct best vectors, selected through tournament selection criteria will be helpful in balancing the exploration and exploitation. To minimize the loss diversity a tournament of small size is selected so that less fit individuals may have a chance of selection as a parent. TSDE generates two best vectors using tournament selection mechanism. The selection of parents is based on the tournament selection criteria by taking a tournament of size 3 of randomly selected population members. The equation of the proposed variant TSDE is as follows

$$v_g^i = x_g^{r1} + F(x_g^{best1} - x_g^{r1}) + F(x_g^{best2} - x_g^{r2}) \qquad \text{..................(5.1)}$$

The equation (5.1) of this variant uses two best vectors x_g^{best1} and x_g^{best2} and two random vectors x_g^{r1} and x_g^{r2} that are selected from the current population. Vectors x_g^{best1} and x_g^{best2} are selected by using tournament selection mechanism by taking tournament of any size say k. Vector x_g^{best1} is the best individual from first tournament and the vector x_g^{best2} is the second best individual from the second tournament. The vectors used in each tournament are selected randomly from the current population. The Proposed TSDE variant has the ability to increase the convergence speed since it uses the combination of best vectors and random vectors from the population. Tournament selection may incorporate some diversity in TSDE since global best vector has less chance of selection as a parent because tournament of small size is used instead of whole population. Selection of two best individuals instead of global best from the whole population makes it more opportune to escape from the local optima problem that accordingly increases the solution quality and the convergence speed of DE algorithm. The most state of the art DE algorithms focuses on basic DE mutation strategies and the incorporation of this proposed mutation strategy will be helpful in improving the performance of DE algorithm.

A graphical representation of proposed mutation strategy is shown in Figure 5.1, where mutant vector v_g^i is generated using four vectors x_g^{tbest1}, x_g^{tbest2}, x_g^{r1} and x_g^{r2} shown in filled circles and directed arrow shows difference vector.

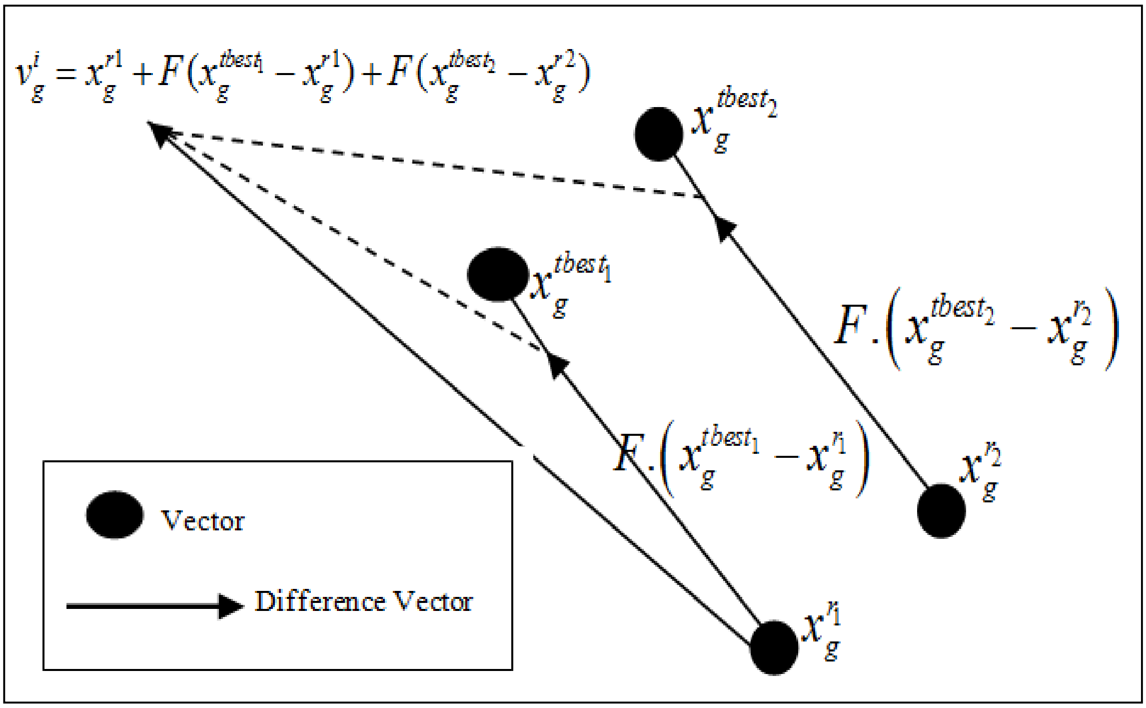


Figure 5.1: Graphical view of proposed TSDE

5.1.2 TSDE Flowchart

Figure 5.2 shows the flowchart of tournament selection base mutation variant of DE algorithm. The flowchart shows various stages where the whole the execution of tournament selection base mutation variant starts the initialization of parameter, NP, F, CR and max_iteration, tournament size and tournament vectors(tbest) size. The evolutionary process contains DE operators and proposed mutation variant tournament selection based criteria. When the stopping criteria finishes we get the optimal solution.

5.1.3 Pseudocode of TSDE

Figure 5.3 shows the pseudocode of tournament selection base mutation variant of DE algorithm. The pseudocode contains an implementation flow of TSDE algorithm that starts with the random initialization of DE population members. After initialization the fitness value of each population member is calculated then TSDE population member are evolved by applying DE algorithm operators mutation, crossover and selection. The tournament selection criteria is used in the selection of parents for proposed mutation strategy. After evolutionary process the optimal solution is obtained. The proposed mutation variant TSDE is also implemented through computer simulation.

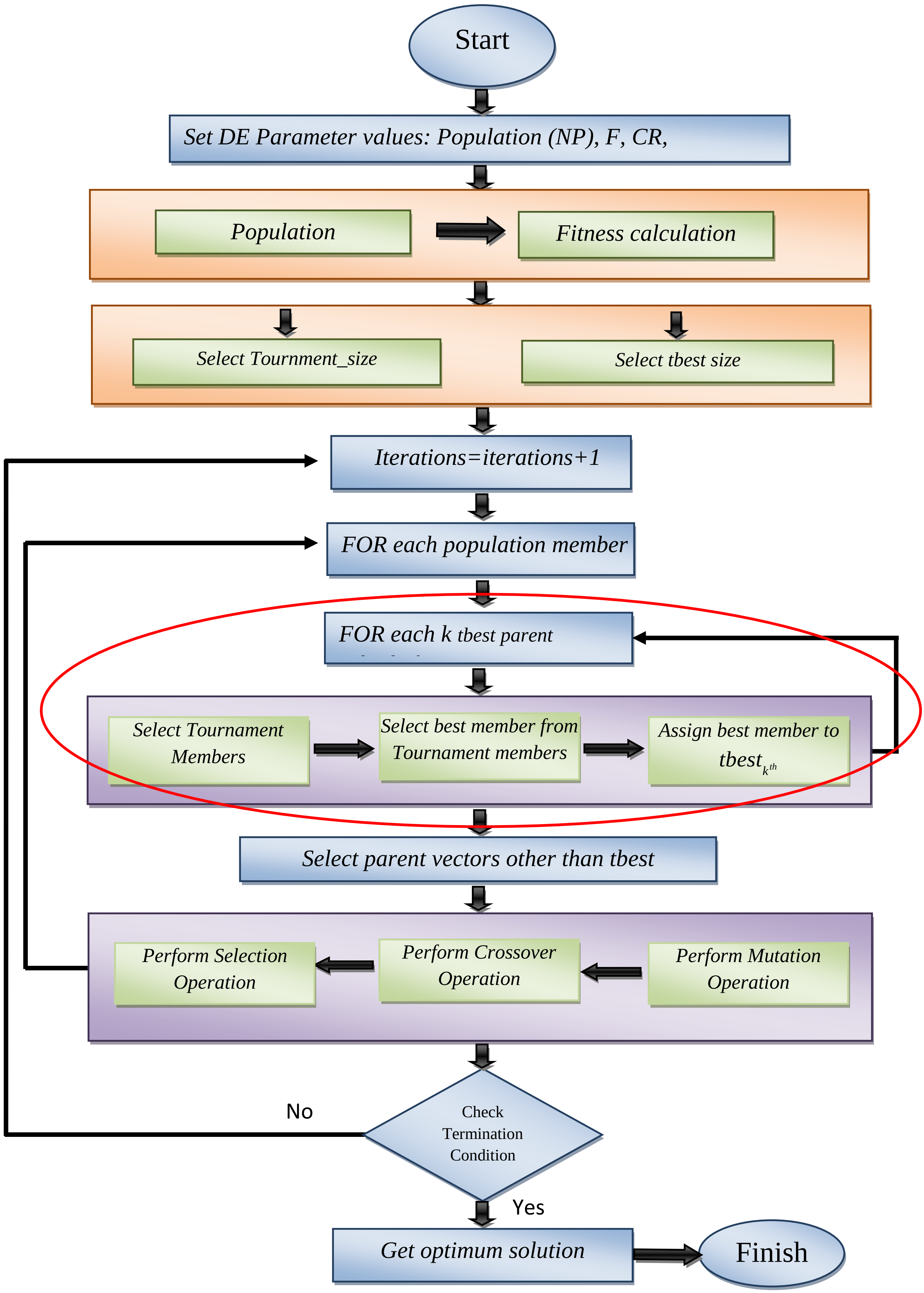


Figure 5.2: Flowchart of tournament selection base mutation variant of DE algorithm

1. Generate the initial population $\mathrm{P_G} = \{\mathrm{X_{1,G}}, \ldots\ldots \mathrm{X_{NP,G}}\}$ for generation G=0 and randomly initialize each population member $\mathrm{X_{i,G}} = \{\mathrm{x}_{i,G}^{1} \ldots\ldots\ldots \mathrm{x}_{i,G}^{D}\}$ where i = 1…………NP

2. FOR i = 1 to NP

Calculate fitness $f(X_{i,G})$ for each population member $\mathrm{X_{i,G}}$

END FOR

3. WHILE the stopping criterion is not true

/ Start of TSDE vector's selection */*

Step 3.1 TSDE vectors selection

FOR n = 1 to number of *TSDE* vectors

FOR k = 1 to *Tournment_size*

Select k^{th} tournament member with its fitness randomly from current population

END FOR

Select best of best member from the current tournament as $\boldsymbol{n^{th}}$ *tbest*

Return $\boldsymbol{n^{th}}$ member index to be used as one of *TSDE* vectors in proposed mutation strategy

END FOR

/ End of TSDE vectors selection */*

Step 3.2 Mutation Step

FOR i = 1 to NP

For the i^{th} target vector $\mathrm{X}_{i,G}$ generate a donor vector $V_{i,G} = \{v_{i,G}^{1} \ldots\ldots v_{i,G}^{D}\}$ with the specified mutation strategy (From Table-I strategies or proposed *TSDE* strategy)

END FOR

Step 3.3 Crossover Step

FOR i = 1 to NP

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

END FOR

Step 3.4 Selection Step

FOR i=1 to NP

Evaluate the trial vector $U_{i,G}$ against the target vector $\mathrm{X}_{i,G}$ with fitness function f

IF $f(U_{i,G} \leq f\left(X_{i,G}\right)$, THEN $X_{i,G+1} = U_{i,G}$, $f(X_{i,G}) = f\left(U_{i,G}\right)$

IF $f(U_{i,G} \leq f\left(X_{best,G}\right)$, THEN $X_{best,G+1} = U_{i,G}$,

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

END IF

END IF

END FOR

Step 3.5 increment generation number G=G+1

END WHILE

4. *Get Optimum individual*

Figure 5.3: *Pseudocode of Tournament Selection based DE algorithm (TSDE)*

5.1.4 Salient features of TSDE

The main features of the proposed mutation strategy are as follows

1) A random number is used as a base vector since mutation strategies *"DE/rand/1"* and *"DE/rand/2"* uses random base vectors that maintains better exploration ability but have slow convergence speed **[**33**]**.
2) Although DE mutation strategies *"DE/best/1"*, *"DE/rand-to-best"* and *"DE/best/2"* have fast convergence [169] but these are biased towards global best value that may have a chance to stuck in local optima problem [33]. So to improve the convergence speed of proposed TSDE, two best vectors ($best_1$, $best_2$) are selected using tournament selection criteria while above mentioned strategies uses one global best vector.
3) The equation of proposed mutation strategy is

$$v_g^i = x_g^{r1} + F(x_g^{best1} - x_g^{r1}) + F(x_g^{best2} - x_g^{r2}) \quad \text{.............................(5.2)}$$

Where F is scaling factor of each difference vector. The motivation of this mutation strategy is an existing DE variant *"DE/rand to best/1"* [98-99] that places perturbation of random vector towards global best vector and contains two difference vectors with an equation

$$v_g^i = x_g^{r1} + F(x_g^{best} - x_g^{r1}) + F(x_g^{r2} - x_g^{r3}) \quad \text{......................(5.3)}$$

Two best vectors ($best_1$, $best_2$) in TSDE will be helpful in escaping local optima problem and be helpful in improving the solution quality.
4) Two difference vectors are used where each difference vector utilizes one best vector selected through tournament criteria. Since mutation strategies having two difference vectors (DV) produce better perturbation mode than one difference vector mutation strategies **[**120**]**.
5) The members of the tournament are selected randomly that almost makes it equal chance of selection of higher or lower performing individuals from the population. This variant maintains randomness since members of the tournament are selected randomly and we select the best performing amongst the tournament to ignore the poor performing individuals. The proposed variation is simple since it uses two best individuals selected using the tournament selection technique and other random vector selected randomly from the current population. The performance TSDE is compared with the other DE variants in the result section.

5.1.5 Benchmark functions and TSDE Parameter Setting

In order to evaluate the performance of the proposed variant and existing variants of DE algorithm, a comprehensive set of 30 N-Dimensional benchmark functions is used. These benchmark functions are commonly used for multidimensional global optimization problems having variety of characteristics. Searching capability of Optimization algorithm can be accessed by applying them to the problem of such diverse characteristics [170]. Modality, Separability and dimensionality are important characteristics of functions to test optimization algorithm[170-72]. Test set of diverse properties given in the appendix section is used to test the effectiveness of the proposed variation of DE algorithm. The functions given in appendix section contain equation, graphs, search space, dimension and optimum value of each function. N-dimensional functions have been used for an extensive comparison of DE algorithm variants for various dimensions. The control parameters of the DE algorithm are N_p (Population Size), D(Dimension), F(Scaling Factor) and CR(Crossover control parameter). Different researchers have used different F and CR parameter values for function optimization of N-dimensional functions like Noman and Iba [173] have used F=0.9, CR=0.9, Breast et al. [116] have used F=0.5, CR=0.9; Ali and Torn [174] have used CR=0.5, $F \in [0.4,1]$; Piotrowski and Napiorkowski [175] have used CR=0.5 and F=0.5 and Dong et al. [176] have used F=0.7 and CR=0.5 in their research work. In this research work we have used control parameters F=0.7, CR=0.5 and population size $N_p = 30$ [177]. The Dimensions size used is 10, 20 and 30 and the number of training iterations for average fitness value are 5000, 10000 and 15000 for 10D, 20D and 30D respectively. The DE algorithm iterates over 30 independent runs and the average fitness value of these 30 runs is used in the results. Same parameter values are used for all functions and all mutation strategies. Experimental results of the number of function calls (NFC) are generated for maximum NFC 10^4 *DIM [178]. To find out the number of function calls, VTR is set to 0.0001 and Max-NFC for 10D, 20D and 30D is respectively 100,000; 200,000 and 300,000 for all functions and all mutation strategies.

5.1.6 Simulation results of TSDE

This section presents simulation results of proposed TSDE and other mutation strategies of DE algorithm. Furthermore, comparative analysis of TSDE and other DE mutation strategies is also given in this chapter. In order to perform a fair comparison all results are averaged over 30 independent runs using parameter settings given in section 5.1.5. Both Binomial and Exponential crossover schemes are implemented for all benchmark functions given in the appendix and DE mutation strategies given in table-4.2 to obtain the results on a set of benchmark functions for the set of DE mutation strategies. Both Average fitness value and the number of calls (NFC) are used for performance evaluation of DE mutation strategies as used by most of the evolutionary computing algorithms **[**169**]**. First, average fitness values of the functions are considered over 30 independent runs for the performance evaluation of DE mutation strategies. For easy observation best values of mutation strategies are given as bold face in every case. Experimental results of average fitness for all DE mutation strategies and benchmark functions are reported in table 5.1. Because of space limitation the results of binomial and exponential strategies of the proposed TSDE, five most commonly used strategies **[**169**]** and another one of the best performing strategy are reported in table-5.1-2. The average fitness Dendrograms for the proposed DE mutation variant and existing DE mutation strategies are given in Figures 5.22-29. Convergence graphs of binomial and exponential versions of the proposed TSDE and other selected mutation strategies are shown in Figure 5.4-21. NFC experimental results for DE mutation strategies and benchmark functions are reported in table 5.2.

The results of average fitness value and NFC are used to evaluate the performance of DE mutation strategies. Commonly used mutation strategies, proposed mutation variant and one another better performing mutation strategy results are included in the result section since results are large enough to manage in the tabular form for all variants in the thesis. Experimental results are obtained for 40 DE mutation strategies given in table-4.2 and the proposed TSDE mutation variant (V_{41} and V_{42}) but reported for selected mutation strategies only due to space issue. Experimental results are analyzed on the basis of commonly used performance parameters average fitness and NFC. The benchmark functions are N-dimensional functions having varying nature.

From average fitness results it can be observed that proposed binomial TSDE (V_{41}) version has dominating performance among all strategies; the mutation strategy V_{10} that has never announced to be a one of the best performing mutation strategy in DE has the second best performance; proposed exponential TSDE (V_{42}) and two other well known DE mutation strategies $V_{7,}$ V_2 have subsequent best performance. Although in most of the cases proposed V_{41} shares its best performance with other mutation strategies like V_{10}, V_7, V_2, etc but convergence graphs shows that V_{41} has better convergence speed than these mutation strategies. Research result shows that proposed mutation strategies perform better for the functions having various characteristics like unimodal/multimodal, separable/non-separable. The proposed mutation strategy (V_{41}) has better average fitness performance for separable functions f_1, f_2, f_{12}, f_{13}, f_{19}*(10D,20D)*, f_{16}*(10D)*, f_{22}*(10D,20D)*, $f_{18,}$ f_{19}, f_{20}, f_{21}, f_{22}*(10D)*, f_{26}, f_{28};

non-separable functions f_3, f_7, f_{10}, f_{11}, f_{18}, f_{24}, f_{25}; uni-modal functions f_2, f_3, f_{11}, f_{12}, f_{13}, f_{16}*(10D)*, f_{26}, f_{21} and multimodal functions f_1, f_7, f_{10}, f_{18}, f_{19}*(10D,20D)*, f_{22}*(10D,20D)*, f_{18}, f_{19}, f_{22}*(10D)*, f_{24}, f_{25}, f_{26}, f_{28}. One of the rarely used DE mutation strategy V_{10} has second best average fitness performance for separable functions f_1, f_2, f_{12}*(10D)*, f_{14}, f_{19}, f_{16}*(10D)*, f_{18}, f_{24}, f_{20}, f_{21}, f_{26}; non-separable functions f_7, f_{18}, f_{21}*(10D,30D)*; uni-modal functions f_2, f_{12}*(10D)*, f_{14}, f_{16}*(10D)*, f_{20}, f_{21} and multimodal functions f_1, f_7, f_{18}, f_{19}, f_{21}*(10D,30D)*, f_{18}, f_{24}, f_{26}. The proposed exponential mutation strategy (V_{42}) has better average fitness performance for separable functions f_5 *(10D,30D)*,f_{12}, f_{14}, f_{22}, f_{22}*(10D)*, f_{26}; non-separable functions f_6*(30D)*, f_{24}, f_{25}, f_{30}; uni-modal functions, f_{12}, f_{14} and multimodal functions f_5*(10D,30D)*,f_6*(30D)*, f_{22}, f_{22}*(10D)*, f_{24}, f_{25}, f_{26}, f_{30}. Average fitness result shows that V_{41} has dominating performance among all strategies and, although it shares its best performance with other mutation strategies like V_2, V_7, V_{10} in most of the cases, but the convergence graphs shows that the proposed TSDE has fast convergence than other mutation strategies like V_2, V_7, V_{10} and vice versa.

Results of number of function call's shows that TSDE(V_{41}) has clearly leading performance among all other DE mutation strategies in most of the cases. The binomial TSDE mutation variant V_{41} has best performance in most of the functions having various characteristics like separable, non-separable, uni-modal and multimodal. V_{41} outperforms other strategies for *Sphere model*, *Axis parallel hyperellipsoid*, *Schwefel's problem 1.2*, *Rosenbrock's valley*, *Griewangk's function (20D,30D)*,*Sum of different power*, *Levy function (30D)*, *Zakharov function*, *Schwefel's problem 2.22*, *Step function*, *De Jong's function-4*, *Levy and Montalvo Problem*, *Cosine Mixture*, *Cigar*, *Function '15'*, *Ellipse Function*, *Tablet Function*, *Schewel*, *MultiModal global optimization problem*, *Quintic global optimization problem* and *Stochastic global optimization problem*. The exponential version of TSDE mutation variant V_{42} has better performance for *Ackley's path function (30D), Levy function (20D)* and *Alpine function(20D)*. One of the commonly used mutation strategy of DE algorithm V_2 has better performance for *Griewank's function (10D)*, *Mishra-1 global optimization problem*, *Mishra-2 global optimization problem*, *Stretched-V global optimization problem(30D)* and *XinSheYang(30D)* functions. The mutation strategies that fail to reach to VTR for any function with specified parameter contains dashed values in table-5.2 for the NFC results. It can noted from the results that the proposed TSDE performs better than all others including V10. Proposed TSDE can be one of the powerful mutation strategy in DE algorithm.

Experimental result can be summarized that the proposed binomial *"TSDE (V_{41})"* has the dominating performance among all DE mutation strategies for both average fitness values and number of function call values. The strategy *V_{10} ("DE/rand repeated to best/1/bin")* is the second best performing mutation strategy in average fitness value and is runner up in most of the cases of number of function call values. One interesting aspect of this variant (V_{10}) is that the performance of this strategy is never announced to be one the best strategy of the DE algorithm. The mutation strategy V_7 and V_2*(“DE/best/1/bin”)* that are already known to be best performing DE variant have subsequent best performance along with exponential *“TSDE* *(*V_{42}*)”* and vice versa. It cans also be summarized that the worst performing mutation strategies among the considered strategies are V_6 *(“DE/rand repeating & current to rand /1/bin”),* V_8, V_{11}*(DE/rand to best/1),* V_{14}*(“DE/rand to best/2”)* V_{16}*(“DE/rand to current /2”),* V_{19} *(“DE/rand/3”),* V_{20}*(“DE/best/3”)* and vice versa.

Table.5.1 Fitness Results (Mean ± S.D) of DE variants for functions (f_1- f_{13})(achieves optimum value in iterations)

Fun	DIM	Iter	v1	v2	v3	v4	v7	v10	v41	v42
f_1	10D	5000	1.76E-145±7.33E-145	2.69e-310±0.00E+00	1.53E-69±3.05E-69	1.09E-114±3.68E-114	2.47e-315±0.00E+00	**0.00E+00±0.00E+00 (4700 iterations)**	**0.00E+00±0.00E+00 (4210 iterations)**	2.94E-255±0.00E+00
	20D	10000	3.25E-109±7.70E-109	2.22e-315±0.00E+00	9.92E-30±1.71E-29	1.64E-63±2.68E-63	**0.00E+00±0.00E+00 (4670 iterations)**	**0.00E+00±0.00E+00 (4240 iterations)**	**0.00E+00±0.00E+00 (3500 iterations)**	9.86E-235±0.00E+00
	30D	15000	2.10E-80±7.19E-80	1.24e-313±0.00E+00	1.95E-12±1.18E-12	6.81E-38±9.97E-38	**0.00E+00±0.00E+00 (4240 iterations)**	**0.00E+00±0.00E+00 (4000 iterations)**	**0.00E+00±0.00E+00 (3210 iterations)**	1.71E-227±0.00E+00
f_2	10D	5000	1.03E-165±0.00E+00	**0.00E+00±0.00E+00 (4530 iterations)**	3.53E-84±6.08E-84	9.30E-135±2.20E-134	**0.00E+00±0.00E+00 (4530 iterations)**	**0.00E+00±0.00E+00 (4330 iterations)**	**0.00E+00±0.00E+00 (3880 iterations)**	5.95E-270±0.00E+00
	20D	10000	6.33E-118±1.08E-117	**0.00E+00±0.00E+00 (4670 iterations)**	1.85E-33±2.16E-33	1.68E-70±3.09E-70	**0.00E+00±0.00E+00 (4270 iterations)**	**0.00E+00±0.00E+00 (4060 iterations)**	**0.00E+00±0.00E+00 (3530 iterations)**	2.28E-240±0.00E+00
	30D	15000	3.65E-85±6.96E-85	9.88e-324±0.00E+00	3.23E-13±2.21E-13	4.26E-40±1.11E-39	**0.00E+00±0.00E+00 (4140 iterations)**	**0.00E+00±0.00E+00 (3910 iterations)**	**0.00E+00±0.00E+00 (3140 iterations)**	1.44E-229±0.00E+00
f_3	10D	5000	2.01E-29±3.29E-29	1.39E-86±4.53E-86	3.33E-11±3.84E-11	3.37E-22±5.88E-22	2.70E-94±8.76E-94	1.03E-86±5.42E-86	**9.92E-120±4.77E-118**	4.73E-58±3.59E-58
	20D	10000	4.71E-01±6.18E-01	6.41E-26±1.32E-25	8.11E+02±2.40E+02	3.69E+01±2.20E+01	4.88E-34±1.79E-33	1.78E-28±3.51E-28	**1.36E-47±6.90E-47**	7.22E-24±1.15E-23
	30D	15000	2.19E+03±7.20E+02	4.16E-10±6.03E-10	7.93E+03±9.36E+02	4.70E+03±1.04E+03	4.45E-16±7.71E-16	1.06E-11±1.79E-11	**4.00E-26±6.90E-26**	1.50E-14±1.67E-14
f_4	10D	5000	4.07E-11±7.30E-11	5.32E-01±1.36E+00	8.29E-04±1.26E-03	2.89E-08±4.09E-08	4.51E-01±1.21E+00	2.66E-01±9.94E-01	2.66E-01±7.16E-01	**6.91E-16±7.16E-01**
	20D	10000	**5.89E-06±6.91E-06**	6.64E-01±1.49E+00	7.63E+00±6.43E-01	2.36E+00±1.24E+01	1.33E-01±7.16E-01	2.66E-01±9.94E-01	1.33E-01±7.16E-01	6.97E-01±1.21E+00
	30D	15000	2.66E-02±1.20E-02	7.97E-01±1.59E+00	3.45E+01±2.89E+01	9.00E+00±7.07E-01	3.99E-01±1.20E+00	3.99E-01±1.20E+00	3.99E-01±1.20E+00	9.34E+00±3.05E+00
f_5	10D	5000	6.63E-02±2.48E-01	4.15E+00±1.74E+00	2.67E+00±2.67E+00	1.76E+00±1.14E+00	2.18E+00±6.27E-01	9.95E-01±1.06E+00	1.53E+00±1.17E+00	3.32E-02±0.00E+00
	20D	10000	3.24E+01±6.67E+00	1.18E+01±3.55E+00	6.01E+01±6.13E+00	5.37E+01±5.16E+00	2.50E+01±5.62E+00	4.34E+00±1.35E+00	6.33E+00±2.03E+00	6.63E-02±2.48E-01
	30D	15000	1.02E+02±6.58E+00	2.34E+01±6.93E+00	1.48E+02±1.06E+01	1.32E+02±6.88E+00	7.02E+01±9.81E+00	1.00E+01±2.93E+00	1.34E+01±3.64E+00	1.66E-01±3.71E-01
f_6	10D	5000	3.20E-03±6.67E-03	4.96E-02±2.41E-02	1.68E-01±3.13E-02	4.70E-02±3.52E-02	1.76E-02±1.17E-02	2.08E-02±1.59E-02	1.44E-02±1.53E-02	9.86E-04±5.95E-03
	20D	10000	1.07E-03±2.75E-03	1.01E-02±1.53E-02	3.53E-02±7.05E-02	8.46E-03±8.64E-03	2.87E-03±5.65E-03	2.71E-03±4.51E-03	2.96E-03±4.82E-03	**0.00E+00±0.00E+00**
	30D	15000	4.93E-04±1.84E-03	5.82E-03±9.68E-03	8.32E-02±7.92E-02	4.93E-03±5.36E-03	1.07E-03±2.75E-03	2.30E-03±4.71E-03	4.93E-04±1.84E-03	**0.00E+00±0.00E+00**
f_7	10D	5000	2.59E-195±0.00E+00	**0.00E+00±0.00E+00 (3830 iterations)**	7.97E-92±1.43E-91	2.61E-143±1.06E-142	**0.00E+00±0.00E+00 (3800 iterations)**	**0.00E+00±0.00E+00 (3570 iterations)**	**0.00E+00±0.00E+00 (2990 iterations)**	8.00E-294±0.00E+00
	20D	10000	6.79E-164±0.00E+00	**0.00E+00±0.00E+00 (3100 iterations)**	1.64E-40±6.56E-40	4.66E-84±2.39E-83	**0.00E+00±0.00E+00 (3060 iterations)**	**0.00E+00±0.00E+00 (2840 iterations)**	**0.00E+00±0.00E+00 (2340 iterations)**	1.65E-299±0.00E+00
	30D	15000	2.01E-111±5.48E-111	**0.00E+00±0.00E+00 (2980 iterations)**	7.97E-17±9.77E-17	1.28E-49±4.28E-49	**0.00E+00±0.00E+00 (2920 iterations)**	**0.00E+00±0.00E+00 (2740 iterations)**	**0.00E+00±0.00E+00 (2260 iterations)**	2.10E-300±0.00E+00
f_8	10D	5000	1.25E-02±3.76E-02	2.56E-01±8.15E-02	3.36E-01±8.74E-02	1.26E-01±8.21E-02	1.61E-01±4.84E-02	7.33E-02±8.17E-02	9.79E-02±1.01E-01	4.18E-03±5.21E-13
	20D	10000	7.30E-01±7.45E-02	3.51E-01±7.27E-02	1.07E+00±7.52E-02	9.67E-01±6.48E-02	5.40E-01±9.69E-02	1.55E-01±6.27E-02	2.32E-01±6.28E-02	8.85E-03±2.66E-02
	30D	15000	1.09E+00±7.43E-02	3.52E-01±6.25E-02	1.57E+00±6.60E-02	1.34E+00±8.02E-02	8.14E-01±1.04E-01	2.26E-01±6.83E-02	2.73E-01±5.21E-02	7.22E-03±2.16E-02
f_9	10D	5000	3.88E-07±4.11E-07	2.66E-07±2.48E-07	1.02E-06±8.02E-07	7.74E-07±5.97E-07	4.14E-07±3.93E-07	3.49E-07±3.89E-07	3.31E-07±3.83E-07	1.97E-08±1.68E-08
	20D	10000	3.60E-07±3.74E-07	1.39E-07±1.38E-07	1.10E-06±7.89E-07	5.86E-07±6.51E-07	1.57E-07±1.46E-07	1.55E-07±1.24E-07	1.55E-07±1.44E-07	4.36E-09±5.29E-09
	30D	15000	2.23E-07±1.98E-07	1.03E-07±7.96E-08	1.10E-06±8.75E-07	5.59E-07±4.14E-07	7.24E-08±6.20E-08	3.16E-02±1.70E-01	1.86E-03±1.00E-02	1.42E-09±1.24E-09
f_{10}	10D	5000	9.21E-43±2.10E-42	9.04E-114±2.78E-113	3.46E-17±3.08E-17	1.10E-31±2.27E-31	3.32E-121±1.75E-120	1.16E-117±3.13E-117	**2.11E-151±5.43E-151**	1.71E-37±1.06E-36
	20D	10000	1.68E-09±1.15E-09	7.49E-44±1.88E-43	1.29E+00±3.74E-01	2.46E-04±1.36E-04	1.29E-53±3.18E-53	3.71E-50±1.09E-49	**3.83E-71±6.60E-71**	5.69E-02±4.26E-02
	30D	15000	4.80E-02±2.97E-02	2.26E-24±3.92E-24	4.92E+01±6.73E+00	8.27E+00±2.25E+00	1.39E-32±2.11E-32	5.45E-30±9.59E-30	**1.11E-44±4.40E-44**	2.08E+00±7.83E-01
f_{11}	10D	5000	3.69E-80±8.46E-80	2.82E-165±0.00E+00	4.43E-38±3.82E-38	1.26E-61±2.12E-61	2.58E-162±1.48e-323	3.46E-171±0.00E+00	**7.33E-202±0.00E+00**	5.07E-134±6.53E-134
	20D	10000	5.37E-62±7.70E-62	5.49E-170±0.00E+00	8.19E-17±5.19E-17	3.89E-35±3.50E-35	1.88E-179±0.00E+00	3.80E-190±0.00E+00	**1.67E-232±0.00E+00**	1.94E-124±2.30E-124
	30D	15000	9.57E-47±1.12E-46	2.46E-171±0.00E+00	6.24E-07±2.38E-07	5.79E-21±6.51E-21	1.12E-188±0.00E+00	2.16E-202±0.00E+00	**1.60E-257±0.00E+00**	3.84E-120±3.69E-120
f_{12}	10D	5000	**0.00E+00±0.00E+00 (170 iterations)**	**0.00E+00±0.00E+00 (90 iterations)**	**0.00E+00±0.00E+00 (330 iterations)**	**0.00E+00±0.00E+00 (220 iterations)**	**0.00E+00±0.00E+00 (100 iterations)**	**0.00E+00±0.00E+00 (80 iterations)**	**0.00E+00±0.00E+00 (70 iterations)**	**0.00E+00±0.00E+00 (100 iterations)**
	20D	10000	**0.00E+00±0.00E+00 (230 iterations)**	1.00E-01±3.00E-01	**0.00E+00±0.00E+00 (830 iterations)**	**0.00E+00±0.00E+00 (420 iterations)**	3.33E-02±1.80E-01	3.33E-02±1.80E-01	**0.00E+00±0.00E+00 (60 iterations)**	**0.00E+00±0.00E+00 (110 iterations)**
	30D	15000	**0.00E+00±0.00E+00 (320 iterations)**	6.67E-01±1.01E+00	**0.00E+00±0.00E+00 (2050 iterations)**	**0.00E+00±0.00E+00 (760 iterations)**	6.67E-02±2.49E-01	3.33E-02±1.80E-01	**0.00E+00±0.00E+00 (100 iterations)**	**0.00E+00±0.00E+00 (120 iterations)**
f_{13}	10D	5000	2.56E-285±0.00E+00	**0.00E+00±0.00E+00 (2500 iterations)**	6.35E-143±1.79E-142	5.39E-231±0.00E+00	**0.00E+00±0.00E+00 (2480 iterations)**	**0.00E+00±0.00E+00 (2400 iterations)**	**0.00E+00±0.00E+00 (2030 iterations)**	**0.00E+00±0.00E+00 (3010 iterations)**
	20D	10000	7.71E-177±0.00E+00	**0.00E+00±0.00E+00 (2480 iterations)**	9.82E-49±1.53E-48	5.74E-104±1.53E-103	**0.00E+00±0.00E+00 (2520 iterations)**	**0.00E+00±0.00E+00 (2390 iterations)**	**0.00E+00±0.00E+00 (1980 iterations)**	**0.00E+00±0.00E+00 (3390 iterations)**
	30D	15000	4.51E-114±2.31E-113	**0.00E+00±0.00E+00 (3070 iterations)**	4.50E-17±4.05E-17	1.10E-53±3.40E-53	**0.00E+00±0.00E+00 (2610 iterations)**	**0.00E+00±0.00E+00 (2410 iterations)**	**0.00E+00±0.00E+00 (2020 iterations)**	**0.00E+00±0.00E+00 (3570 iterations)**
f_{14}	10D	5000	4.65E-05±5.05E-05	4.72E-05±4.27E-05	5.14E-05±4.25E-05	6.14E-05±5.44E-05	5.24E-05±4.40E-05	4.91E-05±4.00E-05	3.39E-05±5.08E-05	1.35E-05±1.52E-05
	20D	10000	3.07E-05±3.23E-05	1.80E-05±1.35E-05	2.77E-05±2.74E-05	4.63E-05±4.66E-05	2.59E-05±2.30E-05	2.52E-05±1.89E-05	3.05E-05±3.73E-05	5.24E-06±5.41E-06
	30D	15000	3.44E-05±3.54E-05	1.67E-05±1.50E-05	4.24E-05±4.56E-05	2.51E-05±3.07E-05	2.44E-05±2.35E-05	1.91E-05±1.74E-05	1.79E-05±1.77E-05	**2.01E-06±1.29E-06**
f_{15}	10D	5000	**3.27E-31±1.31E-46 (1220 iterations)**	**3.27E-31±1.31E-46 (600 iterations)**	**3.27E-31±1.31E-46 (2520 iterations)**	**3.27E-31±1.31E-46 (1560 iterations)**	**3.27E-31±1.31E-46 (920 iterations)**	**3.27E-31±1.31E-46 (540 iterations)**	**3.27E-31±1.31E-46 (470 iterations)**	**3.27E-31±1.31E-46 (700 iterations)**
	20D	10000	**1.63E-31±6.57E-47 (1660 iterations)**	5.18E-03±2.79E-02	1.17E-28±1.56E-28	**1.63E-31±6.57E-47 (2880 iterations)**	**1.63E-31±6.57E-47 (3660 iterations)**	**1.63E-31±6.57E-47 (500 iterations)**	**1.63E-31±6.57E-47 (410 iterations)**	**1.63E-31±6.57E-47 (750 iterations)**
	30D	15000	**1.09E-31±8.76E-47 (2240 iterations)**	6.91E-03±2.59E-02	3.43E-10±5.64E-10	**1.09E-31±8.76E-47 (4800 iterations)**	**1.09E-31±8.16E-35 (1070 iterations)**	**1.09E-31±1.38E-34 (470 iterations)**	3.46E-03±1.86E-02	**1.09E-31±8.76E-47 (790 iterations)**

Table.5.1 (Continue) Fitness Results (Mean ± S.D) of DE variants for functions (f_{15}- f_{30})

Fun	DIM	Iter	v_1	v_2	v_3	v_4	v_7	v_{10}	v_{41}	v_{42}
f_{16}	10D	5000	**8.53E+01±7.11E-14 (290 iterations)**	**8.53E+01±6.23E-14 (130 iterations)**	**8.53E+01±6.95E-14 (480 iterations)**	**8.53E+01±7.03E-14 (250 iterations)**	**8.53E+01±5.58E-14 (150 iterations)**	**8.53E+01±5.82E-14 (140 iterations)**	**8.53E+01±7.11E-14 (120 iterations)**	**8.53E+01±7.11E-14 (220 iterations)**
	20D	10000	1.21E-04±1.10E-04	7.97E-05±8.70E-05	1.94E-04±1.87E-04	1.86E-04±1.92E-04	4.77E-05±4.62E-05	3.96E-05±2.79E-05	6.73E-05±7.35E-05	5.62E-06±6.26E-06
	30D	15000	1.49E-04±1.67E-04	1.13E-04±1.14E-04	4.53E-04±5.16E-04	4.49E-04±4.36E-04	6.63E-05±6.82E-05	6.64E-05±6.02E-05	7.43E-05±7.72E-05	9.25E-06±8.13E-06
f_{17}	10D	5000	**0.00E+00±0.00E+00 (620 iterations)**	9.85E-03±3.69E-02	**0.00E+00±0.00E+00 (1380 iterations)**	**0.00E+00±0.00E+00 (830 iterations)**	**0.00E+00±0.00E+00 (590 iterations)**	3.70E-18±1.99E-17	**0.00E+00±0.00E+00 (250 iterations)**	**0.00E+00±0.00E+00 (370 iterations)**
	20D	10000	**0.00E+00±0.00E+00 (840 iterations)**	5.91E-02±8.18E-02	**0.00E+00±0.00E+00 (3190 iterations)**	**0.00E+00±0.00E+00 (1730 iterations)**	9.85E-03±3.69E-02	2.22E-17±6.66E-17	**0.00E+00±0.00E+00 (220 iterations)**	**0.00E+00±0.00E+00 (400 iterations)**
	30D	15000	**0.00E+00±0.00E+00 (1300 iterations)**	1.38E-01±1.32E-01	1.15E-10±1.18E-10	**0.00E+00±0.00E+00 (3960 iterations)**	2.46E-02±5.51E-02	1.97E-02±5.02E-02	1.97E-02±5.02E-02	**0.00E+00±0.00E+00 (450 iterations)**
f_{18}	10D	5000	2.38E-140±7.10E-140	**0.00E+00±0.00E+00 (4990 iterations)**	6.26E-64±9.94E-64	8.59E-109±2.49E-108	**0.00E+00±0.00E+00 (4840 iterations)**	**0.00E+00±0.00E+00 (4730 iterations)**	**0.00E+00±0.00E+00 (4030 iterations)**	7.88E-250±0.00E+00
	20D	10000	6.90E-104±1.09E-103	**0.00E+00±0.00E+00 (4940 iterations)**	4.09E-24±6.53E-24	1.36E-57±3.99E-57	**0.00E+00±0.00E+00 (4470 iterations)**	**0.00E+00±0.00E+00 (4250 iterations)**	**0.00E+00±0.00E+00 (3510 iterations)**	2.86E-230±0.00E+00
	30D	15000	3.95E-75±7.03E-75	**0.00E+00±0.00E+00 (4950 iterations)**	8.54E-07±6.45E-07	1.61E-32±2.77E-32	**0.00E+00±0.00E+00 (4270 iterations)**	**0.00E+00±0.00E+00 (4040 iterations)**	**0.00E+00±0.00E+00 (3210 iterations)**	1.24E-221±0.00E+00
f_{19}	10D	5000	3.99E-146±1.16E-145	**0.00E+00±0.00E+00 (4960 iterations)**	9.32E-70±1.47E-69	1.17E-113±5.13E-113	**0.00E+00±0.00E+00 (4880 iterations)**	**0.00E+00±0.00E+00 (4700 iterations)**	**0.00E+00±0.00E+00 (4010 iterations)**	3.12E-255±0.00E+00
	20D	10000	2.65E-109±8.16E-109	**0.00E+00±0.00E+00 (4900 iterations)**	6.25E-30±7.38E-30	1.53E-63±3.40E-63	**0.00E+00±0.00E+00 (4430 iterations)**	**0.00E+00±0.00E+00 (4220 iterations)**	**0.00E+00±0.00E+00 (3480 iterations)**	8.15E-236±0.00E+00
	30D	15000	8.28E-81±1.39E-80	**0.00E+00±0.00E+00 (4920 iterations)**	1.76E-12±9.70E-13	3.91E-38±5.30E-38	**0.00E+00±0.00E+00 (4240 iterations)**	**0.00E+00±0.00E+00 (4010 iterations)**	**0.00E+00±0.00E+00 (3210 iterations)**	3.37E-227±0.00E+00
f_{20}	10D	5000	1.75E-143±4.10E-143	**0.00E+00±0.00E+00 (4990 iterations)**	3.61E-67±6.53E-67	3.89E-112±9.53E-112	1.51e-312±0.00E+00	**0.00E+00±0.00E+00 (7550 iterations)**	**0.00E+00±0.00E+00 (404 iterations)**	5.58E-253±0.00E+00
	20D	10000	1.13E-106±2.38E-106	2.20e-312±0.00E+00	5.06E-27±7.98E-27	4.83E-61±9.92E-61	**0.00E+00±0.00E+00 (4480 iterations)**	**0.00E+00±0.00E+00 (4290 iterations)**	**0.00E+00±0.00E+00 (3510 iterations)**	3.88E-233±0.00E+00
	30D	15000	4.31E-78±6.35E-78	5.90e-311±0.00E+00	8.05E-10±5.01E-10	1.89E-35±2.98E-35	**0.00E+00±0.00E+00 (4280 iterations)**	**0.00E+00±0.00E+00 (4030 iterations)**	**0.00E+00±0.00E+00 (3230 iterations)**	2.42E-224±0.00E+00
f_{21}	10D	5000	1.79E-143±3.30E-143	4.19E-307±0.00E+00	1.75E-66±3.86E-66	9.24E-112±1.79E-111	3.31e-320±0.00E+00	**0.00E+00±0.00E+00 (4760 iterations)**	**0.00E+00±0.00E+00 (4040 iterations)**	1.18E-252±0.00E+00
	20D	10000	2.36E-106±5.82E-106	3.94e-313±0.00E+00	2.76E-27±2.93E-27	5.25E-61±8.13E-61	**0.00E+00±0.00E+00 (4470 iterations)**	**0.00E+00±0.00E+00 (4270 iterations)**	**0.00E+00±0.00E+00 (3520 iterations)**	3.85E-232±0.00E+00
	30D	15000	2.55E-77±1.09E-76	5.21e-319±0.00E+00	8.43E-10±4.46E-10	2.67E-35±3.14E-35	**0.00E+00±0.00E+00 (4290 iterations)**	**0.00E+00±0.00E+00 (4040 iterations)**	**0.00E+00±0.00E+00 (3270 iterations)**	5.27E-224±0.00E+00
f_{22}	10D	5000	**0.00E+00±0.00E+00 (1610 iterations)**	**0.00E+00±0.00E+00 (720 iterations)**	**0.00E+00±0.00E+00 (3570 iterations)**	**0.00E+00±0.00E+00 (2120 iterations)**	**0.00E+00±0.00E+00 (690iterations)**	8.22E-33±4.43E-32	**0.00E+00±3.38E-32 (570 iterations)**	**0.00E+00±0.00E+00 (1130 iterations)**
	20D	10000	8.63E-33±4.65E-32	1.89E-32±6.35E-32	5.70E-18±7.17E-18	**0.00E+00±0.00E+00**	2.34E-31±3.59E-31	1.01E-31±1.32E-31	1.51E-31±1.56E-31	5.11E-31±5.29E-31
	30D	15000	1.40E-31±1.84E-31	2.20E-31±3.06E-31	5.37E-05±4.44E-05	1.58E-27±2.62E-27	1.29E-30±2.89E-30	5.06E-31±3.40E-31	6.86E-31±4.01E-31	1.06E-22±3.32E-22
f_{23}	10D	5000	2.53E-06±2.32E-06	2.67E-06±2.58E-06	5.26E-06±4.46E-06	5.69E-06±6.10E-06	1.24E-06±1.03E-06	1.76E-06±2.20E-06	1.56E-06±1.32E-06	2.88E-07±1.91E-07
	20D	10000	2.94E-06±2.94E-06	1.92E-06±1.80E-06	7.99E-06±6.28E-06	5.69E-06±5.00E-06	1.34E-06±1.45E-06	9.49E-07±1.16E-06	1.57E-06±1.36E-06	**7.14E-08±7.03E-08**
	30D	15000	3.23E-06±2.88E-06	1.69E-06±1.71E-06	1.48E-05±1.48E-05	6.77E-06±5.84E-06	1.24E-06±1.37E-06	1.02E-06±1.04E-06	1.36E-06±1.30E-06	4.64E-08±5.37E-08
f_{24}	10D	5000	**0.00E+00±0.00E+00 (540 iterations)**	7.40E-16±1.66E-15	**0.00E+00±0.00E+00 (800 iterations)**	**0.00E+00±0.00E+00 (430 iterations)**	2.96E-16±1.11E-15	4.44E-16±1.33E-15	**0.00E+00±0.00E+00 (280 iterations)**	**0.00E+00±0.00E+00 (530 iterations)**
	20D	10000	**0.00E+00±0.00E+00 (530 iterations)**	6.19E-15±5.31E-15	**0.00E+00±0.00E+00 (1040 iterations)**	**0.00E+00±0.00E+00 (440 iterations)**	5.34E-15±5.10E-15	3.66E-15±4.71E-15	**0.00E+00±0.00E+00 (250 iterations)**	**0.00E+00±0.00E+00 (610 iterations)**
	30D	15000	**0.00E+00±0.00E+00 (530 iterations)**	1.00E+09±0.00E+00	**0.00E+00±0.00E+00 (1060 iterations)**	1.00E+09±0.00E+00	1.00E+09±0.00E+00	1.00E+09±0.00E+00	**0.00E+00±0.00E+00 (240 iterations)**	**0.00E+00±0.00E+00 (610 iterations)**
f_{25}	10D	5000	**0.00E+00±0.00E+00 (580 iterations)**	2.22E-15±2.22E-15	**0.00E+00±0.00E+00 (910 iterations)**	**0.00E+00±0.00E+00 (560 iterations)**	1.04E-15±1.88E-15	1.63E-15±2.14E-15	**0.00E+00±0.00E+00 (300 iterations)**	**0.00E+00±0.00E+00 (540 iterations)**
	20D	10000	**0.00E+00±0.00E+00 (610 iterations)**	6.19E-15±4.32E-15	**0.00E+00±0.00E+00 (1290 iterations)**	**0.00E+00±0.00E+00 (760 iterations)**	5.06E-15±4.13E-15	5.63E-15±5.03E-15	**0.00E+00±0.00E+00 (360 iterations)**	**0.00E+00±0.00E+00 (710 iterations)**
	30D	15000	**0.00E+00±0.00E+00 (620 iterations)**	1.00E+09±0.00E+00	**0.00E+00±0.00E+00 (1380 iterations)**	1.00E+09±0.00E+00	1.00E+09±0.00E+00	1.00E+09±0.00E+00	**0.00E+00±0.00E+00 (320 iterations)**	**0.00E+00±0.00E+00 (700 iterations)**
f_{26}	10D	5000	**0.00E+00±0.00E+00 (2540 iterations)**	**0.00E+00±0.00E+00 (1250 iterations)**	5.73E-224±0.00E+00	**0.00E+00±0.00E+00 (4920 iterations)**	**0.00E+00±0.00E+00 (1650 iterations)**	**0.00E+00±0.00E+00 (1310 iterations)**	**0.00E+00±0.00E+00 (1080 iterations)**	**0.00E+00±0.00E+00 (1190 iterations)**
	20D	10000	**0.00E+00±0.00E+00 (2510 iterations)**	**0.00E+00±0.00E+00 (740 iterations)**	1.36E-94±7.22E-94	1.10E-199±0.00E+00	**0.00E+00±0.00E+00 (1040 iterations)**	**0.00E+00±0.00E+00 (760 iterations)**	**0.00E+00±0.00E+00 (580 iterations)**	**0.00E+00±0.00E+00 (680 iterations)**
	30D	15000	**0.00E+00±0.00E+00 (2770 iterations)**	**0.00E+00±0.00E+00 (560 iterations)**	3.01E-44±7.05E-44	5.18E-118±2.76E-117	**0.00E+00±0.00E+00 (720 iterations)**	**0.00E+00±0.00E+00 (550 iterations)**	**0.00E+00±0.00E+00 (360 iterations)**	**0.00E+00±0.00E+00 (450 iterations)**
f_{27}	10D	5000	1.97E-15±1.58E-16	**1.30E-15±2.62E-16**	4.96E-08±2.66E-07	1.65E-15±2.27E-16	3.45E-13±1.85E-12	1.57E-15±3.46E-16	1.81E-15±1.98E-16	1.89E-15±2.14E-16
	20D	10000	4.00E-15±1.93E-16	**2.53E-15±4.43E-16**	8.05E+00±1.15E+00	3.41E-07±1.83E-06	2.97E-15±4.67E-16	3.03E-15±4.28E-16	3.42E-15±3.81E-16	3.89E-15±2.19E-16
	30D	15000	6.05E-15±1.49E-16	**3.65E-15±5.32E-16**	5.03E+01±1.37E+01	2.13E+00±4.20E+00	4.23E-15±5.43E-16	4.46E-15±5.77E-16	5.13E-15±2.20E-15	5.86E-15±2.61E-16
f_{28}	10D	5000	7.32E-13±1.08E-12	2.66E-02±7.93E-02	8.10E-01±1.54E-01	8.45E-02±1.30E-01	5.31E-53±2.86E-52	1.98E-28±1.05E-27	**7.35E-118±1.46E-117**	6.82E-49±2.60E-48
	20D	10000	3.80E+00±4.43E-01	9.35E-01±1.26E+00	4.86E+00±5.78E-01	5.48E+00±1.00E+00	1.30E-01±3.36E-01	3.21E-02±1.67E-01	**4.98E-118±1.94E-117**	4.74E-22±5.37E-22
	30D	15000	9.50E+00±7.40E-01	4.88E+00±3.39E+00	1.08E+01±7.54E-01	1.28E+01±1.33E+00	7.84E-01±1.10E+00	6.29E-01±1.46E+00	**1.74E-120±7.93E-120**	3.87E-10±4.29E-10
f_{29}	10D	5000	1.07E-08±2.98E-08	3.12E-07±8.42E-07	8.20E-10±2.05E-09	9.20E-10±1.49E-09	8.71E-10±1.67E-09	9.37E-10±2.28E-09	9.43E-08±7.09E-07	4.86E-08±2.13E-07
	20D	10000	1.58E-08±5.52E-08	1.96E-07±4.95E-07	2.32E-10±3.77E-10	2.08E-10±3.87E-10	**8.01E-11±1.80E-10**	7.44E-10±2.08E-09	1.11E-07±4.98E-07	4.63E-08±8.04E-08
	30D	15000	1.73E-08±7.00E-08	3.01E-07±5.49E-07	5.40E-11±8.01E-11	4.47E-11±5.97E-11	6.90E-11±1.97E-10	2.75E-10±1.03E-09	2.04E-07±9.92E-07	6.45E-08±1.30E-07
f_{30}	10D	5000	**2.08E-04±1.09E-19**	2.18E-04±1.60E-05	3.02E-04±3.11E-05	2.26E-04±2.49E-05	2.13E-04±4.52E-06	2.10E-04±9.85E-06	2.10E-04±4.68E-06	**2.08E-04±2.21E-20**
	20D	10000	1.74E-07±3.77E-08	3.06E-08±5.82E-09	2.67E-07±5.91E-08	2.71E-07±6.91E-08	1.75E-07±4.85E-08	1.04E-07±5.80E-08	2.64E-08±4.35E-09	**1.90E-08±1.48E-24**
	30D	15000	3.34E-10±1.37E-10	5.38E-11±1.00E-10	4.68E-10±2.38E-10	5.80E-10±3.10E-10	3.96E-10±1.40E-10	2.80E-10±1.94E-10	2.54E-12±6.83E-13	**1.29E-12±4.09E-28**

Table.5.2 Number of Function Call's (Mean ± S.D) of DE variants for functions (f_1- f_{13})

Fun	DIM	Iter	v_1	v_2	v_3	v_4	v_7	v_{10}	v_{41}	v_{42}
f_1	10D	10^5	2.17E+02±1.02E+01	1.05E+02±5.36E+00	4.40E+02±1.20E+01	2.73E+02±1.31E+01	1.03E+02±3.88E+00	9.68E+01±3.77E+00	**8.38E+01±4.86E+00**	1.29E+02±5.97E+00
	20D	2x10^5	6.22E+02±1.84E+01	2.18E+02±9.01E+00	2.19E+03±7.94E+01	1.04E+03±3.78E+01	2.01E+02±5.41E+00	1.91E+02±6.81E+00	**1.56E+02±5.21E+00**	2.97E+02±9.27E+00
	30D	3x10^5	1.28E+03±3.04E+01	3.42E+02±1.21E+01	7.56E+03±1.95E+02	2.63E+03±9.07E+01	2.91E+02±7.50E+00	2.77E+02±9.53E+00	**2.21E+02±7.02E+00**	4.80E+02±1.09E+01
f_2	10D	10^5	2.07E+02±9.44E+00	1.04E+02±5.47E+00	4.01E+02±1.95E+01	2.55E+02±1.02E+01	1.03E+02±3.46E+00	9.81E+01±4.21E+00	**8.47E+01±3.34E+00**	1.30E+02±5.28E+00
	20D	2x10^5	6.33E+02±1.53E+01	2.35E+02±1.03E+01	2.10E+03±4.47E+01	1.03E+03±3.19E+01	2.12E+02±4.52E+00	2.03E+02±6.60E+00	**1.69E+02±5.48E+00**	3.29E+02±7.92E+00
	30D	3x10^5	1.36E+03±3.12E+01	3.78E+02±1.32E+01	7.55E+03±2.18E+02	2.76E+03±9.85E+01	3.26E+02±8.83E+00	3.09E+02±8.87E+00	**2.46E+02±7.61E+00**	5.40E+02±1.19E+01
f_3	10D	10^5	1.28E+03±7.67E+01	4.55E+02±3.15E+01	2.97E+03±1.24E+02	1.67E+03±9.23E+01	4.22E+02±3.13E+01	4.56E+02±3.00E+01	**3.30E+02±1.91E+01**	6.73E+02±3.11E+01
	20D	2x10^5	2.00E+04±1.10E+03	3.04E+03±1.60E+02	1.06E+05±5.17E+03	3.74E+04±2.03E+03	2.35E+03±1.28E+02	2.75E+03±1.88E+02	**1.72E+03±1.05E+02**	3.22E+03±1.38E+02
	30D	3x10^5	1.82E+05±1.06E+04	9.64E+03±8.13E+02	-	-	6.96E+03±4.12E+02	8.81E+03±6.54E+02	**4.57E+03±2.43E+02**	7.31E+03±2.20E+02
f_4	10D	10^5	3.08E+03±1.36E+02	1.31E+03±2.23E+02	6.46E+03±3.24E+02	3.67E+03±2.24E+02	2.83E+03±7.80E+03	1.43E+03±6.87E+01	**1.23E+03±5.85E+01**	2.59E+03±3.02E+02
	20D	2x10^5	9.58E+03±2.89E+02	3.17E+03±6.65E+02	3.74E+04±1.53E+03	1.61E+04±7.13E+02	3.24E+03±5.66E+02	3.38E+03±3.04E+02	**2.84E+03±4.03E+02**	1.47E+04±2.95E+03
	30D	3x10^5	2.11E+04±5.34E+02	5.72E+03±8.36E+02	1.54E+05±7.14E+03	4.23E+04±2.36E+03	4.87E+03±1.17E+03	5.59E+03±2.73E+02	**4.78E+03±5.89E+02**	3.87E+04±7.47E+03
f_5	10D	10^5	1.43E+03±2.46E+02	-	5.97E+03±7.61E+02	2.33E+03±2.28E+02	1.93E+04±0.00E+00	7.29E+02±2.13E+02	5.21E+02±1.85E+02	2.88E+02±1.37E+01
	20D	2x10^5	2.09E+04±5.74E+03	-	-	-	-	2.70E+03±0.00E+00	-	6.33E+02±3.50E+01
	30D	3x10^5	1.52E+05±3.90E+04	-	-	-	-	-	-	**1.00E+03±4.41E+01**
f_6	10D	10^5	1.96E+03±4.08E+02	**3.57E+02±9.12E+01**	1.29E+04±2.58E+03	1.99E+03±0.00E+00	2.96E+03±1.48E+03	1.19E+03±0.00E+00	3.92E+02±2.44E+02	5.62E+02±1.01E+02
	20D	2x10^5	1.32E+03±3.11E+02	3.30E+02±1.24E+01	1.12E+04±2.59E+03	2.37E+03±3.45E+02	5.19E+02±2.21E+02	2.88E+02±1.76E+01	**2.36E+02±1.67E+01**	6.02E+02±1.37E+02
	30D	3x10^5	2.09E+03±3.06E+02	4.81E+02±2.38E+01	1.94E+04±3.12E+03	4.37E+03±5.69E+02	5.21E+02±1.85E+02	3.95E+02±2.10E+01	**3.17E+02±1.14E+01**	7.40E+02±5.74E+01
f_7	10D	10^5	1.13E+02±8.69E+00	5.74E+01±4.95E+00	2.28E+02±1.90E+01	1.45E+02±1.23E+01	5.59E+01±3.92E+00	5.46E+01±3.73E+00	**4.50E+01±3.38E+00**	7.39E+01±6.10E+00
	20D	2x10^5	2.62E+02±2.15E+01	9.43E+01±1.01E+01	8.84E+02±6.55E+01	4.39E+02±3.89E+01	8.02E+01±6.20E+00	7.62E+01±4.87E+00	**6.14E+01±4.93E+00**	1.44E+02±1.29E+01
	30D	3x10^5	5.20E+02±4.12E+01	1.24E+02±1.20E+01	2.70E+03±2.99E+02	1.01E+03±9.80E+01	1.03E+02±9.07E+00	9.56E+01±8.35E+00	**7.06E+01±5.21E+00**	2.15E+02±2.95E+01
f_8	10D	10^5	1.93E+03±3.42E+02	-	7.25E+03±1.02E+03	3.31E+03±4.53E+02	2.38E+04±1.11E+04	8.50E+02±1.38E+02	5.53E+02±1.32E+02	3.60E+02±2.12E+01
	20D	2x10^5	2.65E+04±6.98E+03	-	-	1.21E+05±4.57E+04	-	3.69E+03±2.17E+03	-	**7.98E+02±2.34E+01**
	30D	3x10^5	1.96E+05±4.43E+04	-	-	-	-	-	-	**1.26E+03±3.37E+01**
f_9	10D	10^5	4.60E+02±4.40E+02	2.28E+02±1.52E+02	9.09E+02±6.08E+02	6.01E+02±4.11E+02	2.24E+02±1.38E+02	2.64E+02±2.25E+02	2.40E+02±1.66E+02	1.07E+02±3.17E+01
	20D	2x10^5	5.91E+02±2.58E+02	2.49E+02±1.21E+02	2.22E+03±7.05E+02	1.07E+03±4.29E+02	3.13E+02±1.34E+02	2.62E+02±1.56E+02	2.24E+02±1.17E+02	**1.77E+02±1.68E+01**
	30D	3x10^5	9.93E+02±3.39E+02	3.06E+02±9.95E+01	5.29E+03±1.24E+03	2.16E+03±1.05E+03	2.98E+02±1.07E+02	2.57E+02±1.15E+02	**2.53E+02±1.22E+02**	2.64E+02±1.79E+01
f_{10}	10D	10^5	9.25E+02±4.04E+01	3.63E+02±1.99E+01	2.05E+03±1.08E+02	1.17E+03±6.41E+01	3.38E+02±1.41E+01	3.54E+02±2.02E+01	**2.68E+02±1.31E+01**	9.56E+02±6.02E+01
	20D	2x10^5	7.02E+03±2.73E+02	1.92E+03±1.04E+02	2.67E+04±1.01E+03	1.17E+04±5.27E+02	1.61E+03±5.94E+01	1.69E+03±8.28E+01	**1.26E+03±5.44E+01**	4.06E+04±6.37E+03
	30D	3x10^5	2.55E+04±8.58E+02	5.02E+03±2.14E+02	1.66E+05±5.05E+03	5.18E+04±1.75E+03	3.92E+03±1.71E+02	4.25E+03±2.40E+02	**2.96E+03±1.28E+02**	-
f_{11}	10D	10^5	3.97E+02±9.40E+00	1.95E+02±5.49E+00	8.19E+02±2.16E+01	5.20E+02±1.53E+01	2.00E+02±5.59E+00	1.90E+02±5.08E+00	**1.61E+02±4.08E+00**	2.40E+02±6.35E+00
	20D	2x10^5	1.09E+03±2.38E+01	4.03E+02±1.17E+01	3.79E+03±9.29E+01	1.92E+03±6.53E+01	3.88E+02±8.09E+00	3.63E+02±8.34E+00	**2.96E+02±6.82E+00**	5.48E+02±1.18E+01
	30D	3x10^5	2.22E+03±4.22E+01	6.15E+02±1.91E+01	1.27E+04±2.45E+02	4.73E+03±1.08E+02	5.68E+02±1.50E+01	5.28E+02±1.43E+01	**4.17E+02±1.15E+01**	8.78E+02±1.37E+01
f_{12}	10D	10^5	1.48E+02±7.39E+00	7.12E+01±4.46E+00	3.04E+02±2.03E+01	1.87E+02±1.42E+01	6.92E+01±4.36E+00	6.62E+01±3.75E+00	**5.66E+01±5.33E+00**	8.46E+01±5.40E+00
	20D	2x10^5	4.16E+02±1.84E+01	1.51E+02±9.34E+00	1.49E+03±6.56E+01	7.40E+02±4.56E+01	1.44E+02±3.95E+01	1.31E+02±1.32E+01	**1.07E+02±5.02E+00**	1.99E+02±8.18E+00
	30D	3x10^5	8.74E+02±3.96E+01	2.35E+02±1.89E+01	5.56E+03±3.45E+02	1.93E+03±1.33E+02	2.07E+02±1.26E+01	3.80E+02±9.15E+02	**1.50E+02±7.71E+00**	3.24E+02±1.03E+01
f_{13}	10D	10^5	9.34E+01±8.06E+00	4.56E+01±4.36E+00	1.78E+02±1.42E+01	1.13E+02±6.94E+00	4.55E+01±4.43E+00	4.34E+01±2.92E+00	**3.69E+01±4.15E+00**	5.51E+01±3.91E+00
	20D	2x10^5	3.43E+02±1.89E+01	1.18E+02±7.10E+00	1.19E+03±7.86E+01	5.58E+02±4.07E+01	1.05E+02±7.79E+00	9.95E+01±5.08E+00	**8.27E+01±4.63E+00**	1.46E+02±7.33E+00
	30D	3x10^5	8.31E+02±3.93E+01	2.00E+02±1.27E+01	5.18E+03±1.82E+02	1.70E+03±9.52E+01	1.72E+02±7.38E+00	1.60E+02±6.86E+00	**1.29E+02±6.28E+00**	2.49E+02±1.06E+01
f_{14}	10D	10^5	1.82E+04±1.68E+04	1.60E+04±1.27E+04	3.08E+04±2.53E+04	2.39E+04±2.16E+04	1.99E+04±2.03E+04	2.00E+04±1.56E+04	1.80E+04±1.58E+04	5.47E+03±5.38E+03
	20D	2x10^5	3.13E+04±3.06E+04	2.09E+04±1.96E+04	3.54E+04±3.32E+04	4.86E+04±4.94E+04	3.06E+04±2.85E+04	2.85E+04±2.88E+04	1.91E+04±1.98E+04	6.36E+03±6.04E+03
	30D	3x10^5	3.81E+04±3.36E+04	1.88E+04±1.80E+04	4.59E+04±5.15E+04	5.65E+04±5.63E+04	3.03E+04±2.24E+04	2.51E+04±2.15E+04	3.93E+04±2.82E+04	5.28E+03±4.19E+03
f_{15}	10D	10^5	2.22E+02±1.46E+01	1.02E+02±6.34E+00	4.78E+02±2.98E+01	2.90E+02±2.02E+01	1.11E+02±1.25E+01	9.44E+01±5.42E+00	**8.10E+01±5.75E+00**	1.20E+02±6.15E+00
	20D	2x10^5	6.30E+02±2.72E+01	2.08E+02±1.18E+01	2.49E+03±1.37E+02	1.16E+03±8.24E+01	2.02E+02±1.43E+01	1.80E+02±7.48E+00	**1.44E+02±6.32E+00**	2.62E+02±8.76E+00
	30D	3x10^5	1.33E+03±6.54E+01	3.21E+02±2.03E+01	9.71E+03±7.84E+02	3.03E+03±1.97E+02	2.93E+02±2.48E+01	2.57E+02±1.06E+01	**1.98E+02±1.01E+01**	4.04E+02±8.80E+00

Table.5.2 (Continue) Number of Function Call's (Mean ± S.D) of DE variants for functions (f_{14}- f_{26})

Fun	DIM	Iter	V_1	V_2	V_3	V_4	V_7	V_{10}	V_{41}	V_{42}
f_{16}	10D	10^5	-	-	-	-	-	-	-	-
	20D	$2x10^5$	7.01E+04±5.68E+04	4.68E+04±4.65E+04	6.64E+04±5.01E+04	8.44E+04±5.47E+04	4.21E+04±3.51E+04	5.45E+04±5.36E+04	4.42E+04±4.67E+04	5.05E+03±6.35E+03
	30D	$3x10^5$	1.13E+05±8.24E+04	1.00E+05±8.88E+04	1.22E+05±8.32E+04	1.05E+05±7.06E+04	1.06E+05±7.77E+04	9.72E+04±6.93E+04	1.18E+05±7.21E+04	7.72E+03±6.50E+03
f_{17}	10D	10^5	2.19E+02±1.04E+01	1.06E+02±5.23E+00	4.82E+02±2.43E+01	2.92E+02±1.62E+01	1.05E+02±6.10E+00	9.91E+01±4.34E+00	**8.36E+01±3.87E+00**	1.25E+02±6.82E+00
	20D	$2x10^5$	6.44E+02±2.46E+01	2.20E+02±8.57E+00	2.52E+03±1.11E+02	1.17E+03±6.68E+01	2.12E+02±8.93E+00	1.93E+02±8.35E+00	**1.58E+02±7.40E+00**	2.92E+02±8.07E+00
	30D	$3x10^5$	1.38E+03±6.13E+01	3.52E+02±1.24E+01	9.38E+03±4.04E+02	3.06E+03±1.86E+02	3.12E+02±1.24E+01	2.84E+02±1.12E+01	**2.18E+02±7.89E+00**	4.68E+02±1.02E+01
f_{18}	10D	10^5	4.08E+02±1.14E+01	1.93E+02±6.80E+00	8.45E+02±1.82E+01	5.12E+02±1.68E+01	1.90E+02±5.31E+00	1.83E+02±4.36E+00	**1.57E+02±4.75E+00**	2.35E+02±5.47E+00
	20D	$2x10^5$	1.11E+03±2.26E+01	3.93E+02±1.11E+01	3.97E+03±8.17E+01	1.89E+03±5.70E+01	3.57E+02±8.23E+00	3.39E+02±8.29E+00	**2.80E+02±7.64E+00**	5.33E+02±8.07E+00
	30D	$3x10^5$	2.31E+03±4.39E+01	6.02E+02±1.59E+01	1.37E+04±2.86E+02	4.77E+03±1.30E+02	5.22E+02±1.37E+01	4.93E+02±1.10E+01	**3.94E+02±9.44E+00**	8.39E+02±8.88E+00
f_{19}	10D	10^5	2.17E+02±1.03E+01	1.04E+02±4.55E+00	4.40E+02±1.61E+01	2.69E+02±1.37E+01	1.01E+02±4.82E+00	9.74E+01±4.45E+00	**8.26E+01±4.92E+00**	1.24E+02±6.16E+00
	20D	$2x10^5$	6.07E+02±1.66E+01	2.16E+02±7.38E+00	2.12E+03±5.58E+01	1.03E+03±4.26E+01	1.94E+02±7.15E+00	1.87E+02±6.10E+00	**1.54E+02±5.76E+00**	2.95E+02±6.46E+00
	30D	$3x10^5$	1.25E+03±4.71E+01	3.33E+02±1.19E+01	7.44E+03±2.42E+02	2.59E+03±5.35E+01	2.89E+02±9.29E+00	2.73E+02±7.16E+00	**2.18E+02±6.55E+00**	4.76E+02±1.06E+01
f_{20}	10D	10^5	3.07E+02±8.64E+00	1.45E+02±5.27E+00	6.34E+02±1.88E+01	3.84E+02±1.49E+01	1.44E+02±5.10E+00	1.38E+02±5.36E+00	**1.16E+02±5.17E+00**	1.78E+02±6.31E+00
	20D	$2x10^5$	8.41E+02±2.20E+01	2.99E+02±1.23E+01	3.00E+03±7.25E+01	1.43E+03±4.08E+01	2.70E+02±7.63E+00	2.62E+02±6.50E+00	**2.12E+02±7.97E+00**	4.07E+02±1.02E+01
	30D	$3x10^5$	1.74E+03±5.34E+01	4.61E+02±1.76E+01	1.05E+04±1.87E+02	3.59E+03±8.78E+01	4.00E+02±1.09E+01	3.77E+02±7.68E+00	**3.02E+02±7.37E+00**	6.48E+02±1.06E+01
f_{21}	10D	10^5	3.21E+02±1.14E+01	1.54E+02±5.74E+00	6.54E+02±1.50E+01	4.07E+02±1.78E+01	1.54E+02±5.77E+00	1.47E+02±5.88E+00	**1.25E+02±4.85E+00**	1.89E+02±4.21E+00
	20D	$2x10^5$	8.62E+02±1.85E+01	3.10E+02±1.07E+01	3.02E+03±7.26E+01	1.46E+03±4.60E+01	2.80E+02±7.51E+00	2.69E+02±8.40E+00	**2.20E+02±6.22E+00**	4.23E+02±6.09E+00
	30D	$3x10^5$	1.77E+03±2.94E+01	4.75E+02±1.38E+01	1.04E+04±2.29E+02	3.64E+03±1.04E+02	4.11E+02±1.07E+01	3.87E+02±9.58E+00	**3.05E+02±8.38E+00**	6.68E+02±9.36E+00
f_{22}	10D	10^5	3.84E+02±1.48E+01	1.70E+02±7.68E+00	8.29E+02±3.64E+01	4.92E+02±2.60E+01	1.65E+02±5.94E+00	1.63E+02±5.52E+00	**1.37E+02±6.72E+00**	2.43E+02±1.45E+01
	20D	$2x10^5$	1.09E+03±3.17E+01	3.65E+02±1.83E+01	4.29E+03±1.70E+02	1.93E+03±5.76E+01	3.40E+02±1.83E+01	3.33E+02±2.85E+01	**2.98E+02±2.05E+01**	6.37E+02±6.63E+01
	30D	$3x10^5$	2.37E+03±8.71E+01	6.07E+02±3.64E+01	1.60E+04±6.60E+02	4.99E+03±1.97E+02	5.81E+02±4.08E+01	5.49E+02±4.04E+01	**5.07E+02±3.02E+01**	1.10E+03±1.93E+02
f_{23}	10D	10^5	1.54E+03±1.17E+03	1.03E+03±8.73E+02	2.63E+03±2.38E+03	2.13E+03±1.98E+03	9.50E+02±9.39E+02	8.58E+02±1.05E+03	1.27E+03±1.25E+03	2.19E+02±1.61E+02
	20D	$2x10^5$	2.41E+03±1.94E+03	1.97E+03±2.01E+03	8.50E+03±8.82E+03	5.67E+03±4.37E+03	1.64E+03±1.48E+03	1.40E+03±1.53E+03	1.42E+03±1.59E+03	2.28E+02±1.45E+02
	30D	$3x10^5$	3.83E+03±3.61E+03	2.84E+03±2.76E+03	2.62E+04±3.69E+04	1.12E+04±1.11E+04	2.53E+03±2.72E+03	2.34E+03±2.06E+03	2.34E+03±1.89E+03	2.80E+02±1.54E+02
f_{24}	10D	10^5	1.94E+02±6.83E+00	**8.57E+01±3.27E+00**	2.83E+02±8.09E+00	1.44E+02±5.16E+00	1.14E+02±3.75E+00	1.03E+02±5.04E+00	9.76E+01±4.62E+00	1.76E+02±4.43E+00
	20D	$2x10^5$	4.04E+02±9.17E+00	**1.60E+02±4.81E+00**	7.69E+02±2.16E+01	3.33E+02±1.02E+01	2.11E+02±8.69E+00	1.90E+02±7.16E+00	1.83E+02±4.67E+00	4.37E+02±8.56E+00
	30D	$3x10^5$	6.39E+02±1.56E+01	**2.27E+02±6.36E+00**	1.57E+03±3.05E+01	5.74E+02±1.97E+01	2.97E+02±7.23E+00	2.64E+02±7.03E+00	2.60E+02±7.35E+00	7.17E+02±1.31E+01
f_{25}	10D	10^5	2.11E+02±7.63E+00	**9.13E+01±4.25E+00**	3.19E+02±8.31E+00	1.57E+02±6.62E+00	1.22E+02±5.43E+00	1.11E+02±3.99E+00	1.04E+02±3.15E+00	1.86E+02±5.67E+00
	20D	$2x10^5$	4.27E+02±1.29E+01	**1.67E+02±6.41E+00**	8.39E+02±2.20E+01	3.48E+02±1.24E+01	2.20E+02±6.27E+00	1.98E+02±5.03E+00	1.89E+02±4.89E+00	4.47E+02±9.85E+00
	30D	$3x10^5$	6.67E+02±1.29E+01	**2.31E+02±6.24E+00**	1.65E+03±4.70E+01	6.01E+02±2.29E+01	3.03E+02±8.41E+00	2.68E+02±6.62E+00	2.68E+02±6.37E+00	7.28E+02±9.54E+00
f_{26}	10D	10^5	4.15E+01±7.49E+00	2.65E+01±6.34E+00	8.84E+01±2.53E+01	6.42E+01±1.25E+01	2.84E+01±5.12E+00	2.65E+01±3.68E+00	**2.12E+01±4.78E+00**	2.90E+01±4.75E+00
	20D	$2x10^5$	8.44E+01±1.75E+01	4.05E+01±9.54E+00	2.09E+02±6.96E+01	1.39E+02±3.52E+01	5.31E+01±5.60E+01	4.26E+01±2.76E+01	**3.12E+01±1.33E+01**	4.66E+01±7.03E+00
	30D	$3x10^5$	1.46E+02±2.78E+01	5.32E+01±1.02E+01	5.57E+02±1.46E+02	3.45E+02±7.42E+01	5.37E+01±1.13E+01	4.65E+01±7.45E+00	**3.65E+01±5.91E+00**	6.69E+01±6.10E+00
f_{27}	10D	10^5	7.74E+02±7.73E+01	2.63E+02±1.45E+01	3.42E+03±6.64E+02	1.01E+03±1.36E+02	6.25E+02±2.29E+02	2.72E+02±2.20E+01	**2.31E+02±2.05E+01**	3.45E+02±2.14E+01
	20D	$2x10^5$	3.10E+03±4.42E+02	5.39E+02±2.74E+01	4.25E+04±1.16E+04	5.26E+03±8.10E+02	8.94E+02±2.79E+02	5.05E+02±2.77E+01	**4.09E+02±2.27E+01**	7.92E+02±4.47E+01
	30D	$3x10^5$	7.68E+03±1.37E+03	8.10E+02±3.98E+01	2.71E+05±3.28E+04	1.53E+04±2.16E+03	1.08E+03±3.03E+02	7.16E+02±3.18E+01	**5.49E+02±2.28E+01**	1.25E+03±5.70E+01
f_{28}	10D	10^5	2.02E+03±1.97E+02	4.01E+02±5.92E+01	-	2.43E+04±1.39E+04	4.14E+02±3.95E+01	9.85E+02±1.94E+03	2.53E+02±1.42E+01	5.94E+02±2.49E+01
	20D	$2x10^5$	-	9.21E+02±1.05E+02	-	-	1.62E+03±1.45E+03	8.08E+03±2.38E+04	**5.12E+02±1.81E+01**	2.68E+03±1.19E+02
	30D	$3x10^5$	-	-	-	-	9.55E+03±2.29E+04	2.55E+04±6.30E+04	**7.79E+02±2.87E+01**	8.54E+03±5.10E+02
f_{29}	10D	10^5	3.16E+01±3.55E+01	2.60E+01±1.96E+01	4.70E+01±4.33E+01	3.81E+01±4.75E+01	3.27E+01±3.60E+01	3.93E+01±4.95E+01	3.42E+01±3.39E+01	7.96E+01±9.05E+01
	20D	$2x10^5$	**2.01E+01±1.80E+01**	3.34E+01±3.89E+01	3.43E+01±3.33E+01	3.09E+01±3.12E+01	3.14E+01±2.40E+01	3.12E+01±3.38E+01	3.45E+01±3.05E+01	1.80E+02±2.73E+02
	30D	$3x10^5$	3.46E+01±2.25E+01	**1.86E+01±1.66E+01**	2.84E+01±2.43E+01	2.61E+01±2.97E+01	4.05E+01±3.80E+01	2.87E+01±2.63E+01	3.05E+01±2.44E+01	2.24E+02±2.44E+02
f_{30}	10D	10^5	-	-	-	-	-	-	-	-
	20D	$2x10^5$	1.08E+02±6.26E+01	7.48E+01±5.54E+01	9.94E+01±6.90E+01	1.47E+02±6.36E+01	9.35E+01±5.22E+01	7.79E+01±3.93E+01	6.68E+01±2.77E+01	3.11E+01±8.45E+00
	30D	$3x10^5$	1.12E+01±5.90E+00	**1.06E+01±4.91E+00**	1.41E+01±6.35E+00	1.08E+01±5.40E+00	1.09E+01±5.38E+00	1.17E+01±6.30E+00	1.12E+01±6.17E+00	1.34E+01±5.17E+00

5.1.7 TSDE Convergence Graphs

The convergence graphs of selected benchmark functions for 10-dimensions (10D) are given in Figure-5.4-21. Logarithmic convergence graphs of average fitness shows number of iterations horizontally and performance vertically. Horizontal iterations are shown as a multiple of 100. Figures-5.4-21 contains the convergence graphs of five commonly used DE mutation strategies (*"DE/rand/1/bin"(V_1)* ,*"DE/best/1/bin"(V_2)* ,*"DE/rand/2/bin"(V_3)* , *"DE/best/2/bin"(V_4), "DE/current to best/1/bin"(V_7)*), the proposed mutation variants (*"TSDE/bin" (V41), "TSDE/exp" (V_{42})*) and another one the best performing variants *"DE/rand repeated to best/1/bin"(V_{10})*. Convergence graphs of few selected functions are given in the thesis. Most of the convergence graphs shows that the proposed *"TSDE/bin" (V41)* mutation variant has better performance in most of the cases and proposed *"TSDE/exp" (V_{42})* has better performance in few cases. The mutation strategy *"DE/rand repeated to best/1/bin"(V_{10})* proves itself to be another better performing DE strategy, it is following *"TSDE/bin" (V41)* in most cases of convergence graphs. It is important to mention here that *"DE/rand repeated to best/1/bin"(V_{10})* strategy has never been declared to be one of the best performing variant or among top performing variants of DE algorithm. The convergence graph also depicts the dominating performance of the proposed DE mutation variant *"TSDE/bin" (V41)*. Most of the researchers have concentrated on very few basic DE strategies like *"DE/rand/1/bin (V_1)", "DE/best/1/bin (V_2), "DE/rand/2/bin (V_3)", "DE/best/2/bin (V_4)"*, etc and some key performing mutation strategies remained unnoticed in the DE research. This study will prove to be fruitful for DE researchers having choice of different variants for scientific and real time applications.

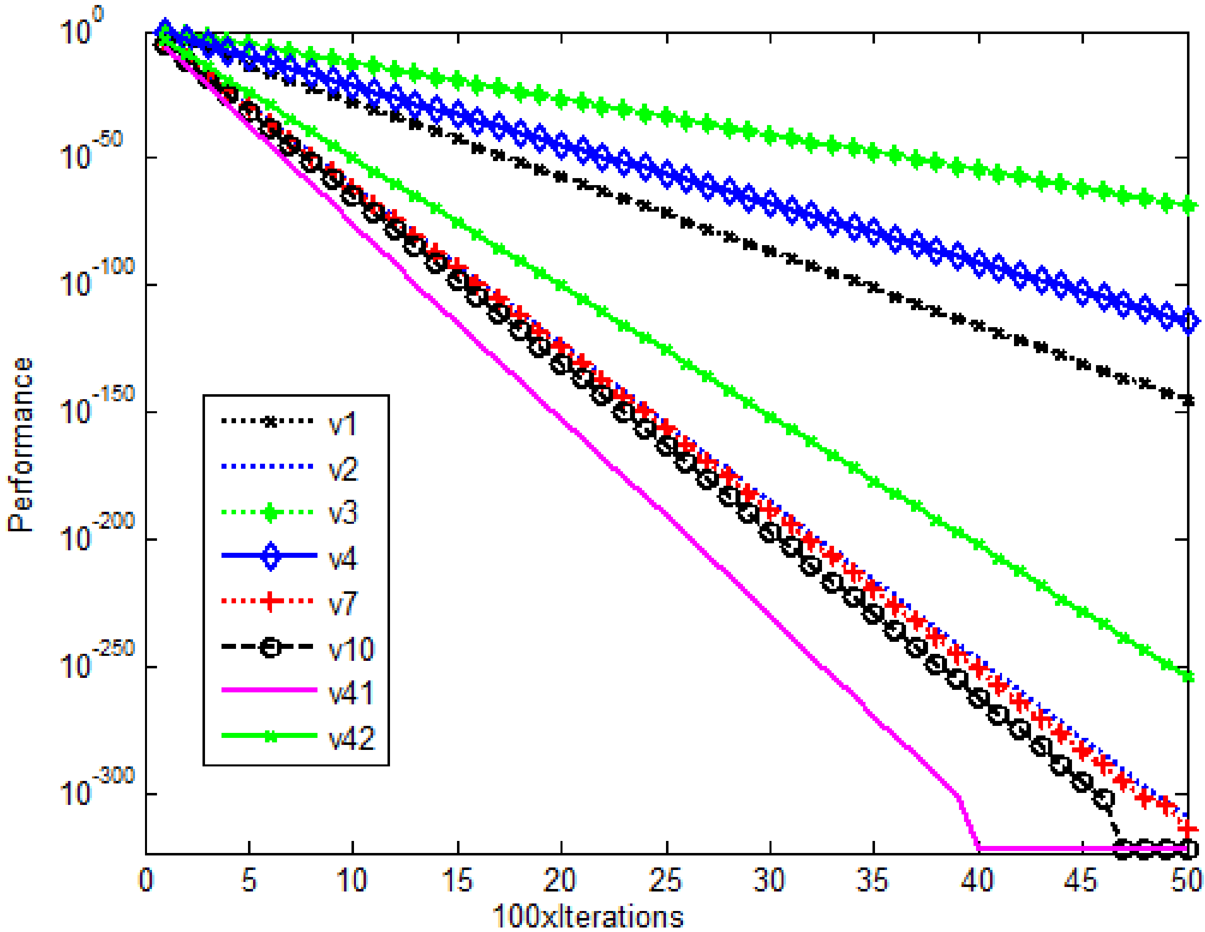


Figure 5.4- 10D average fitness logarithmic convergence graphs for f_1 showing number of iterations horizontally and average fitness vertically

Figure 5.4 presents the average fitness with respect to number of iterations of proposed and conventional mutation strategies on sphere model function. It is obvious from this figure that the performance of *"TSDE/bin" (V41)* is best among all mutation strategies as its fitness value reaches to 0 in 4010 iterations. Among all other techniques *"DE/rand repeated to best/1/bin"(V_{10})* also reaches to 0, however; it takes more iterations than *"TSDE/bin" (V41)*. The important point to note is that the performance of *"DE/rand/2/bin"(V_3)* is worst than all other mutation strategies illustrated in this figure.

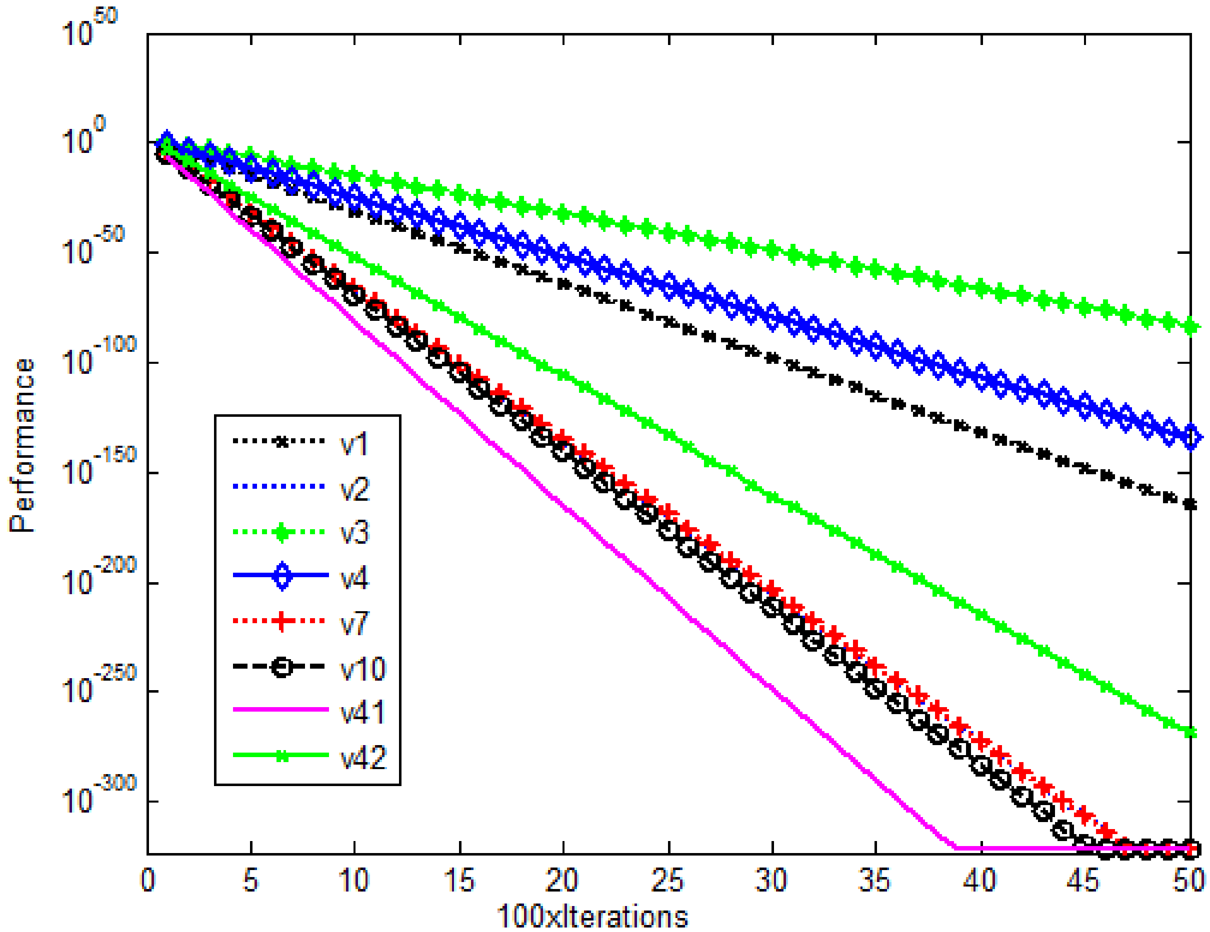


Figure 5.5- 10D average fitness logarithmic convergence graphs for f_2 showing number of iterations horizontally and average fitness vertically

The average fitness of proposed *"TSDE/bin" (V41)* & other traditional mutation strategies for Axis parallel hyperellipsoid (f_2) optimization problem are shown in Figure 5.5. From this figure it is observed that *"TSDE/bin" (V41)*, *"DE/rand repeated to best/1/bin"(V_{10})*, *"DE/current to best/1/bin"(V_7)* and *"DE/best/1/bin"(V_2)* reaches at value 0 within the specified number of iterations, however; *"TSDE/bin" (V41)* achieves value 0 in 3880 iteration while *"DE/rand repeated to best/1/bin"(V_{10})*, *"DE/current to best/1/bin"(V_7)* & *"DE/best/1/bin"(V_2)* takes 4460, 4530 and 4530 iterations respectively. Performance of *"DE/rand/2/bin"(V_3)* is worst among all mutation strategies presented in figure 5.5. It is important to note that performance of *"DE/rand repeated to best/1/bin"(V_{10})*, *"DE/best/1/bin"(V_2)* and *"DE/current to best/1/bin"(V_7)* remains similar in early iterations while it decreases gradually in the succeeding iterations.

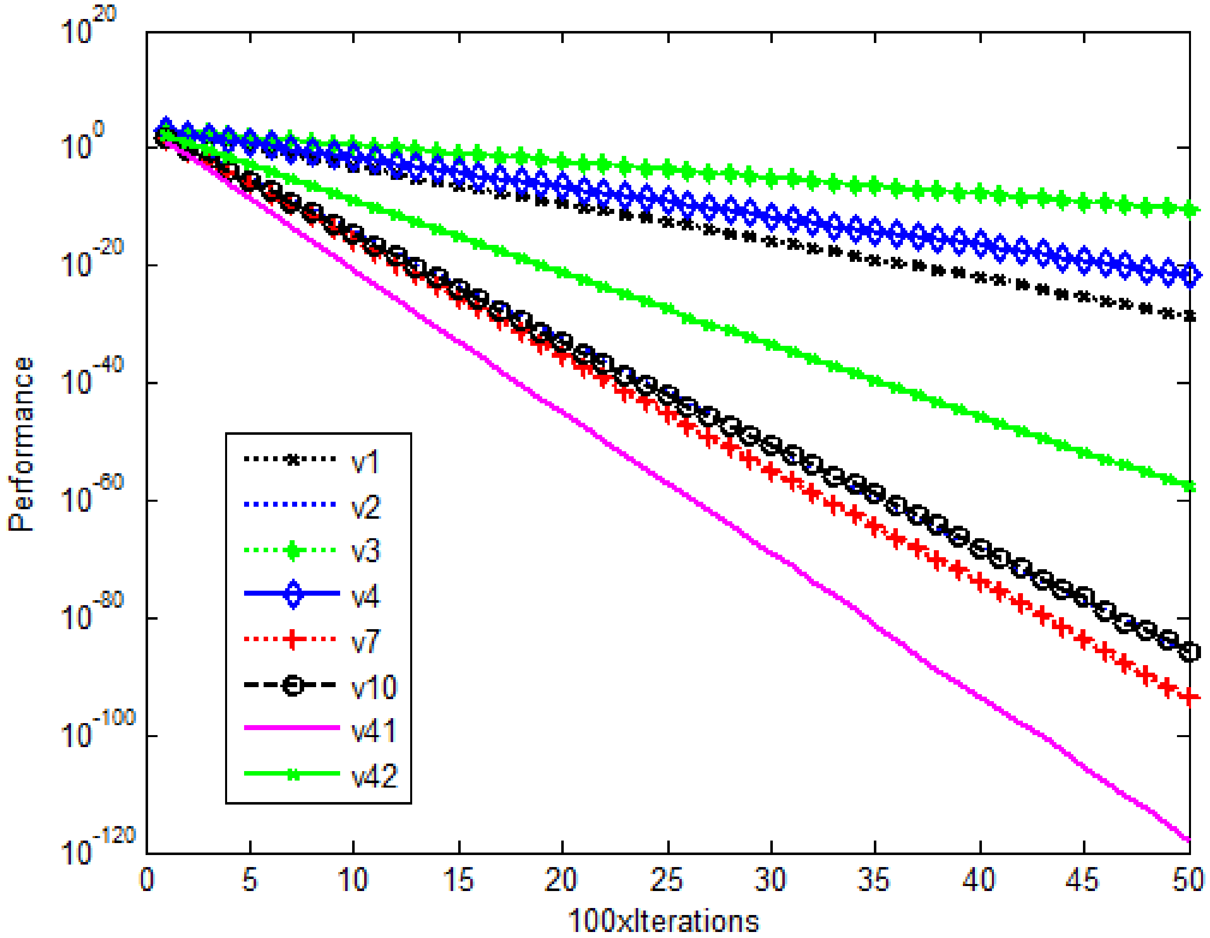


Figure 5.6- 10D average fitness logarithmic convergence graphs for f_3 showing number of iterations horizontally and average fitness vertically

Figure 5.6 illustrates average fitness of *"TSDE/bin" (V41)* and other mutation strategies for Schwefel's problem-1.2 (f_3). This figure depicts that *"DE/rand/2/bin"(V_3)* has worst performance among all mutation strategies. However *"TSDE/bin" (V41)* is quick among all mutation strategies from starting iterations till final iteration. The performance of *"DE/best/1/bin"(V_2)* & *"DE/current to best/1/bin"(V_7)* is similar in almost all iterations while *"DE/rand repeated to best/1/bin"(V_{10})* decreases slowly than *"DE/best/1/bin"(V_2)* & *"DE/current to best/1/bin"(V_7).*

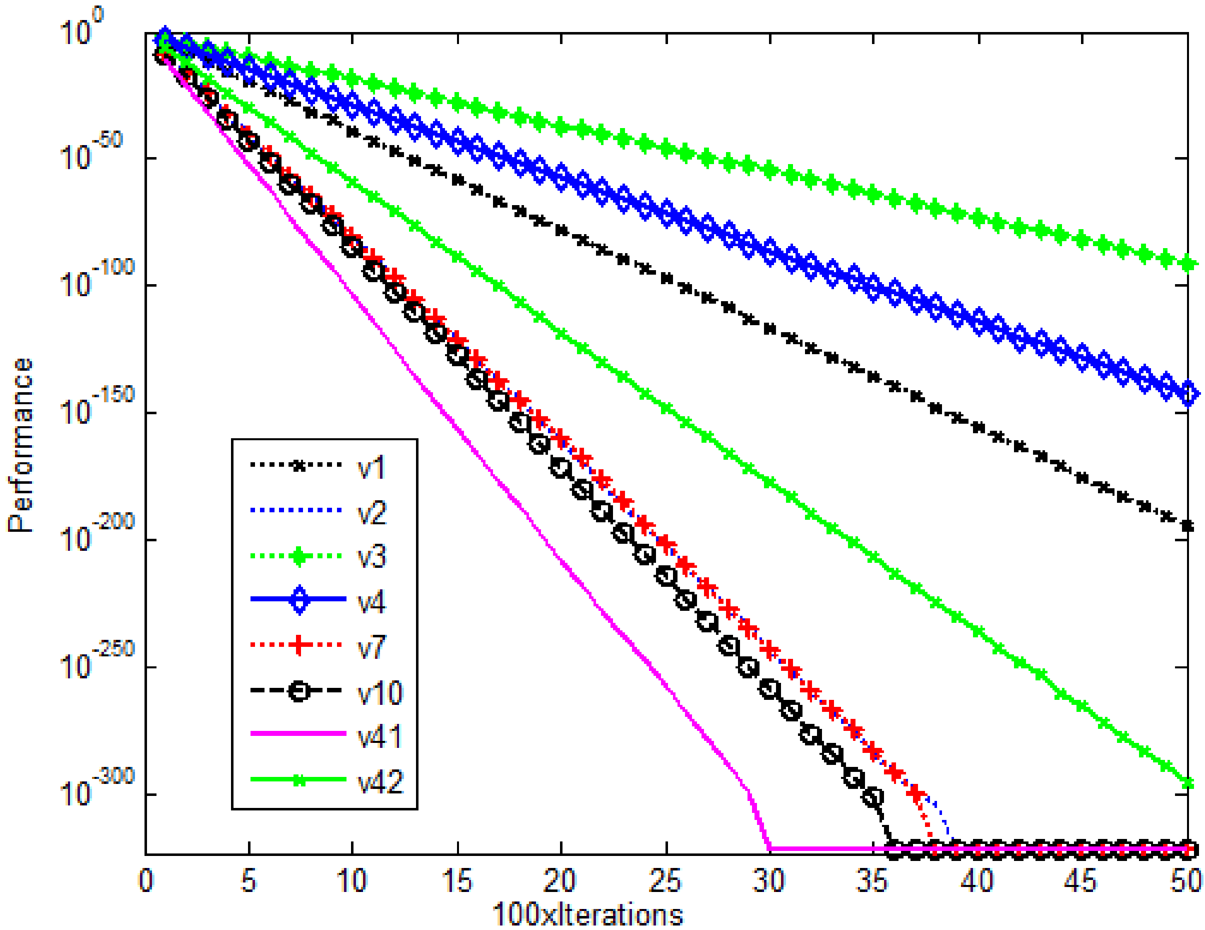


Figure 5.7- 10D average fitness logarithmic convergence graphs for f_7 showing number of iterations horizontally and average fitness vertically

Performance in terms of average fitness value of *“TSDE/bin” (V41)* and all other traditional approaches for Sum of different power function (f_7) is shown in Figure 5.7. It is observed from this figure that performance of *"DE/best/2/bin"(V_4)* is worst among all presented approaches as it cannot reach the minimum value in all 5000 iterations. The fitness value of *“TSDE/bin” (V41), "DE/rand repeated to best/1/bin"(V_{10}), "DE/current to best/1/bin"(V_7), "DE/best/1/bin"(V_2)* reaches to 0 however, among all these the fastest is *“TSDE/bin” (V41)* which achieves 0 fitness value in 2990 iterations while *"DE/rand repeated to best/1/bin"(V_{10}), "DE/current to best/1/bin"(V_7), "DE/best/1/bin"(V_2)* achieves 0 values in 3570, 3800, 3830 iterations respectively.

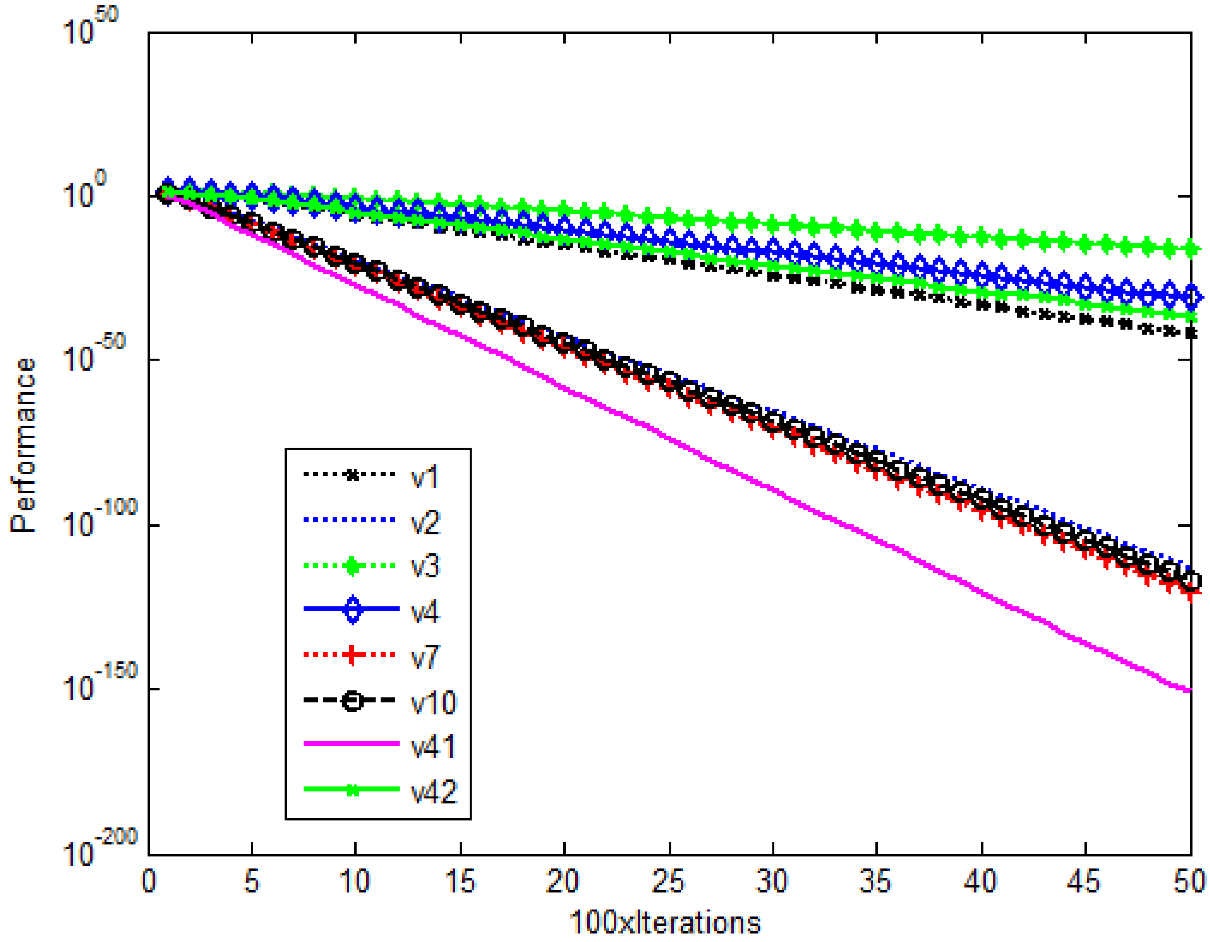


Figure 5.8- 10D average fitness logarithmic convergence graphs for f_{10} showing number of iterations horizontally and average fitness vertically

Figure 5.8 represents the average fitness performance of *"TSDE/bin" (V41)* & other mutation strategies for Zakharov function (f_{10}). This figure depicts that performance of *"DE/current to best/1/bin"(V_7), "DE/rand repeated to best/1/bin"(V_{10})* & *"DE/best/1/bin"(V_2)* is almost similar from initial iteration to final iteration however *"DE/rand/2/bin"(V_3)* has the worst performance among all mutation strategies. It is an important point to note that *"TSDE/bin" (V41)* is converging quickly till final iteration. The average fitness value of *"TSDE/bin" (V41)* is continuously decreasing among all mutation strategies.

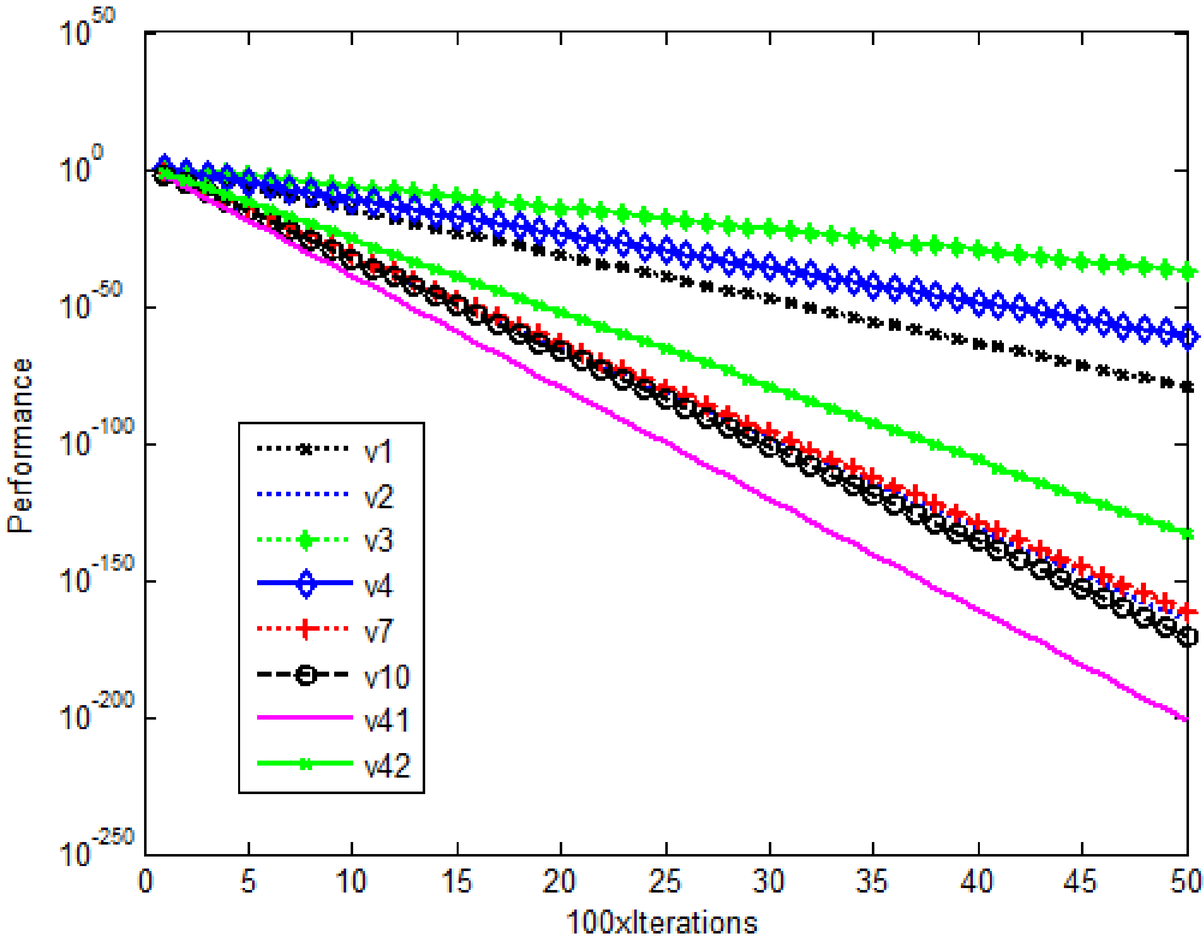


Figure 5.9- 10D average fitness logarithmic convergence graphs for f_{11} showing number of iterations horizontally and average fitness vertically

The performance comparison of *"TSDE/bin" (V41)* and all other commonly used mutation strategies for Schwefel's problem-2.22 (f_{11}) are presented in Figure 5.9. This figure demonstrates that *"DE/best/2/bin"(V_4)* has slowest convergence speed among all other techniques. However the convergence speed of *"TSDE/bin" (V41)* is fastest among all techniques and *"DE/rand repeated to best/1/bin"(V_{10})* has the second best converged. The convergence speed of *"DE/current to best/1/bin"(V_7), "DE/best/1/bin"(V_2)* and *"DE/rand repeated to best/1/bin"(V_{10})* remains similar in early iterations, however; convergence of *"DE/and repeated to best/1/bin"(V_{10})* gradually decreases in subsequent/succeeding iterations. The important point is the performance of *"TSDE/bin" (V41)* is best among all techniques which converge more quickly than all other techniques.

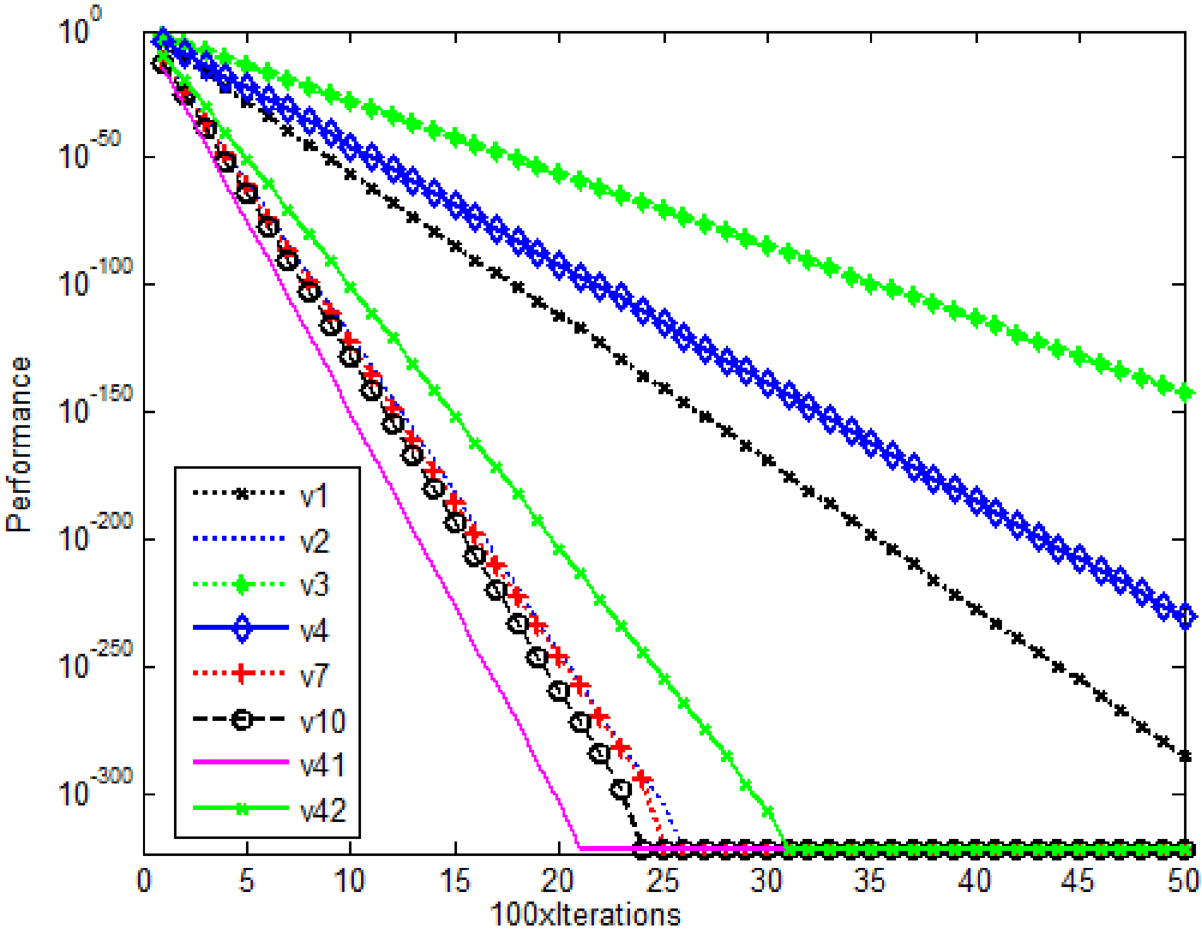


Figure 5.10- 10D average fitness logarithmic convergence graphs for f_{13} showing number of iterations horizontally and average fitness vertically

Figure 5.10 demonstrates the average fitness convergence of *"TSDE/bin" (V41)* and other mutation strategies for De Jong's function-4 (f_{13}). This figure depicts that *"DE/best/2/bin"(V_4)* and *"DE/rand/1/bin"(V_1)* has the worst performance among all other mutation strategies. The fitness value of *"TSDE/bin" (V41)*, *"DE/rand repeated to best/1/bin"(V_{10})*, *"DE/current to best/1/bin"(V_7)*, *"DE/best/1/bin"(V_2)* and *"DE/rand/2/bin"(V_3)* reaches 0 within specified iterations, however; the convergence speed of *"TSDE/bin" (V41)* is best than all other techniques as it fully converges to optimum value in 2030 iterations while *"DE/rand repeated to best/1/bin"(V_{10})*, *"DE/current to best/1/bin"(V_7)* , *"DE/best/1/bin"(V_2)* and *"TSDE/Exp" (V_{42})* takes 2400, 2480, 2500 and 3010 iterations respectively to reach at optimal value. The important point to note is the performance of *"TSDE/bin" (V41)* remains best from early iterations to reach at optimal value.

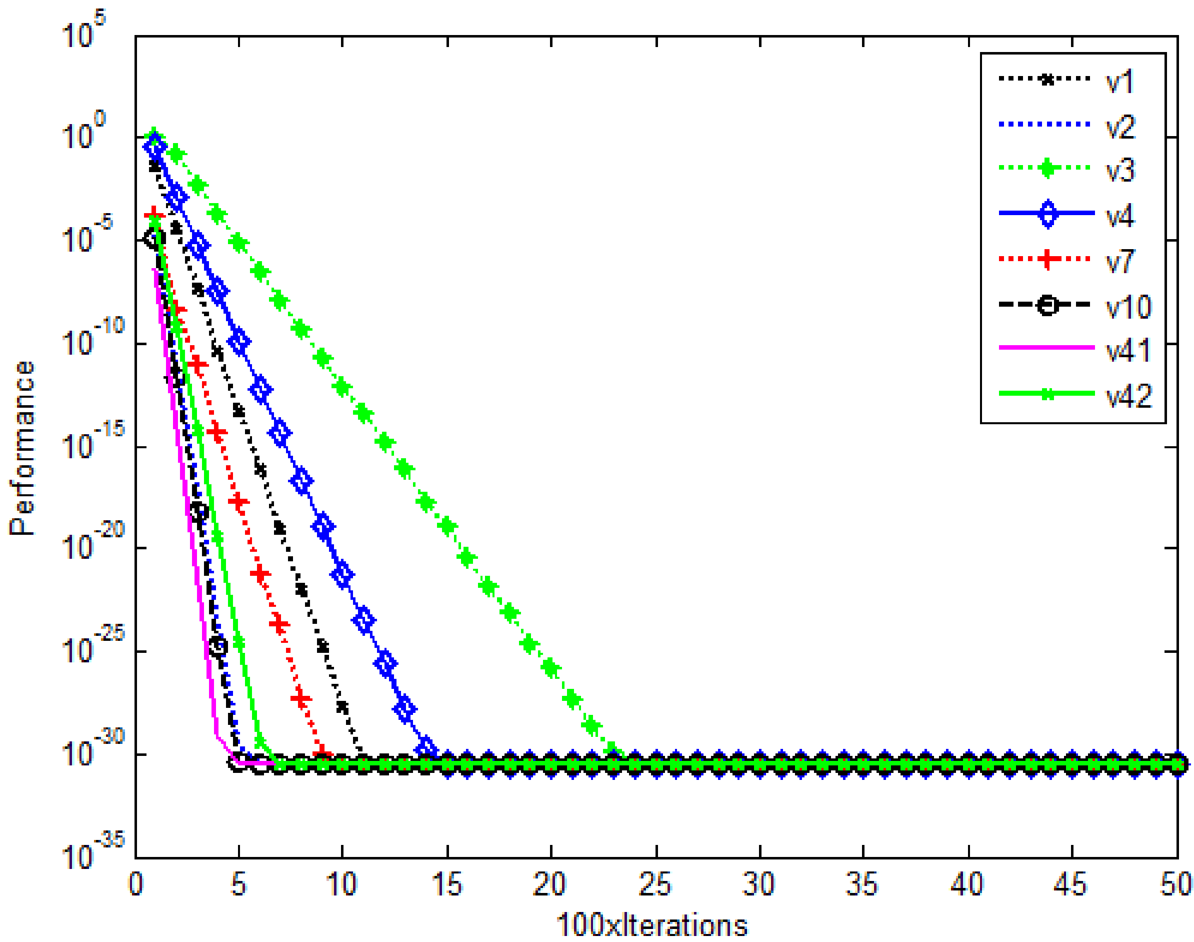


Figure 5.11- 10D average fitness logarithmic convergence graphs for f_{15} showing number of iterations horizontally and average fitness vertically

Figure 5.11 contains the average fitness value of *"TSDE/bin" (V41)*, *"TSDE/Exp" (V_{42})* & other mutation strategies for Levy and Montalvo Problem (f_{15}). This figure illustrates that all mutation strategies attains value 3.27×10^{-31} within the specified iterations. However; *"TSDE/bin" (V41)* is fastest among all mutation strategies that reach at 3.27×10^{-31} in 470 iterations, while *"DE/rand/1/bin"(V_1)*, *"DE/best/1/bin"(V_2)*, *"DE/rand/2/bin"(V_3)*, *"DE/best/2/bin"(V_4)*, *"DE/current to best/1/bin"(V_7)*, *"DE/rand repeated to best/1/bin"(V_{10})* and *"TSDE/Exp" (V_{42})* reaches at this value in 1220, 600, 2520, 1560, 920, 540 and 700 iterations respectively.

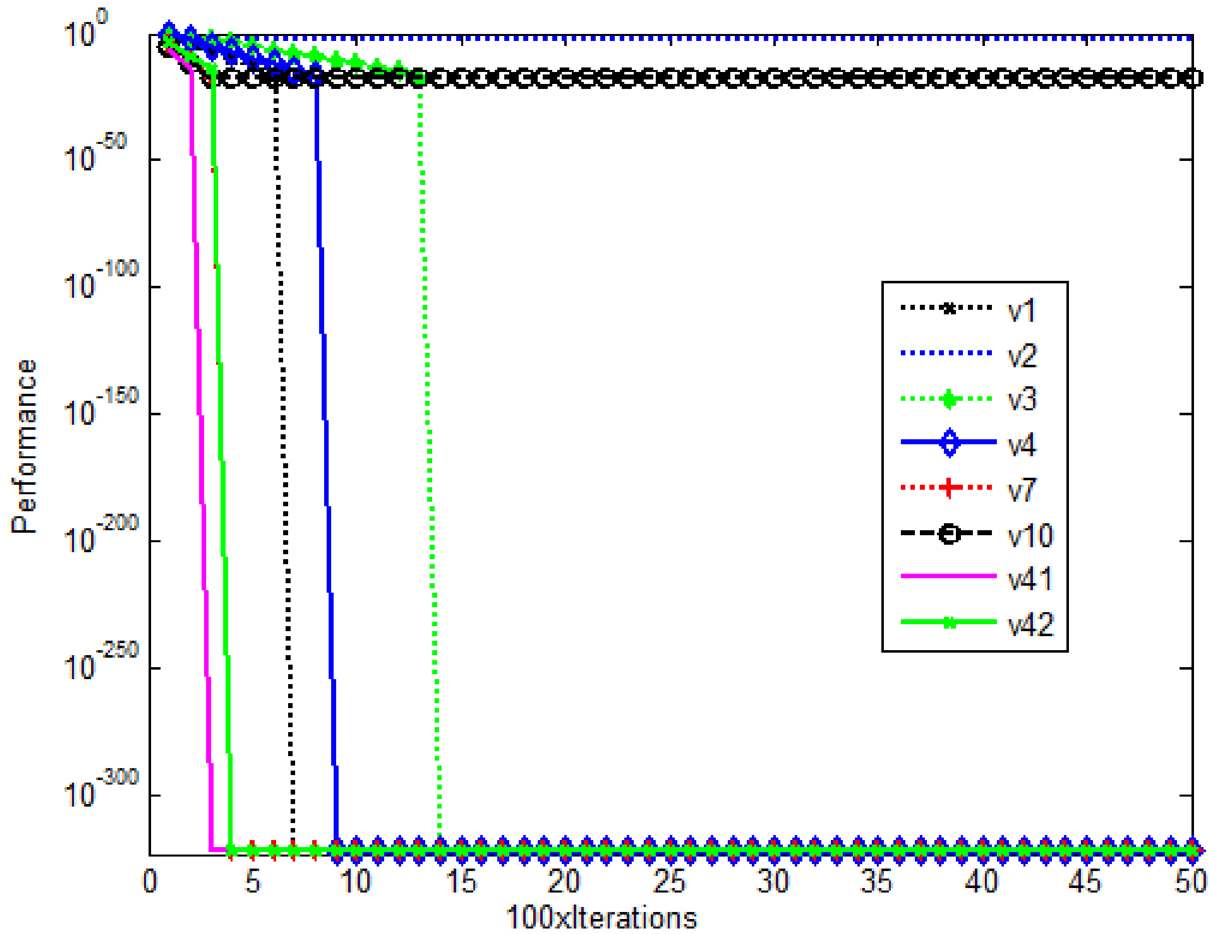


Figure 5.12- 10D average fitness logarithmic convergence graphs for f_{17} showing number of iterations horizontally and average fitness vertically

Figure 5.12 illustrates the average fitness performance of *"TSDE/bin" (V41)* and other mutation strategies for Cosine Mixture function (f_{17}). From this figure, it is noted that the performance of *"DE/best/1/bin"(V_2)* is worst among all strategies that achieves value 9.85×10^{-3} in the specified iterations. Mutation strategies *"TSDE/bin" (V41), "TSDE/Exp" (V_{42}), "DE/current to best/1/bin"(V_7), "DE/rand/1/bin"(V_1), "DE/rand/2/bin"(V_3)* and *"DE/best/2/bin"(V_4)* reached at value 0 in 250, 370, 620, 500, 590, 1380 and 830 iterations respectively. However, it is important to note that *"DE/current to best/1/bin"(V_7)* has quick convergence among all mutation strategies.

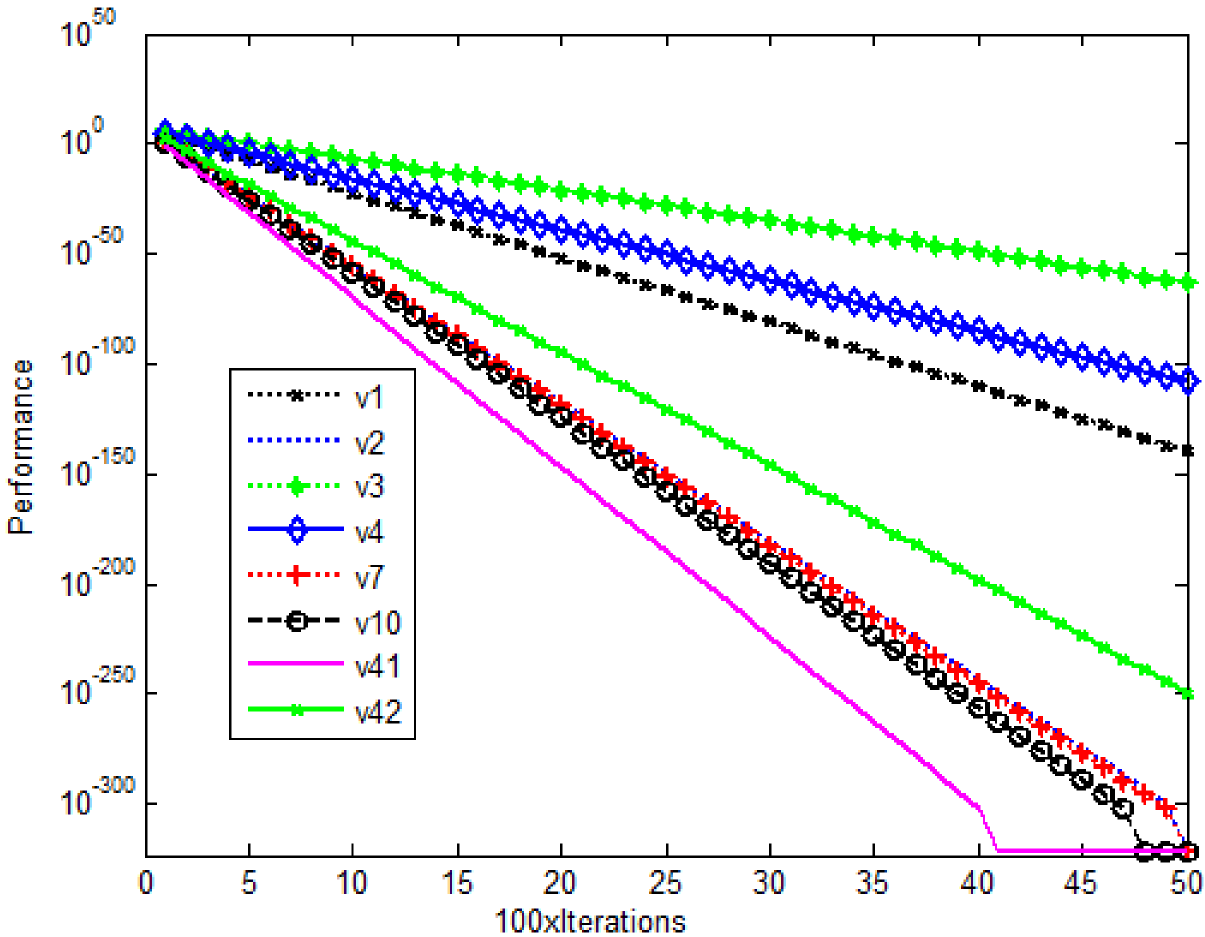


Figure 5.13- 10D average fitness logarithmic convergence graphs for f_{18} showing number of iterations horizontally and average fitness vertically

Average fitness performance of *"TSDE/bin" (V41)* and other mutations strategies is presented in Figure 5.13 for Cigar function (f_{18}). This figure demonstrates that V_3 has the worst performance among all mutation strategies. The mutation strategies *"TSDE/bin" (V41), "DE/rand repeated to best/1/bin"(V_{10}), "DE/current to best/1/bin"(V_7)* and *"DE/best/1/bin"(V_2)* reach at optimal value 0 within a specified number of iterations. However *"TSDE/bin" (V41)* arrives at 0 in 4030 iterations while *"DE/rand repeated to best/1/bin"(V_{10}), "DE/current to best/1/bin"(V_7)* and *"DE/best/1/bin"(V_2)* takes 4730, 4840 and 4990 iterations respectively. From this figure, it is important to note that *"DE/best/1/bin"(V_2)* and *"DE/current to best/1/bin"(V_7)* have an almost similar performance from initial iteration to final iteration. The proposed mutation strategy *"TSDE/bin" (V41)* maintains its quick convergence from starting iterations to reach at 0 among all mutation strategies.

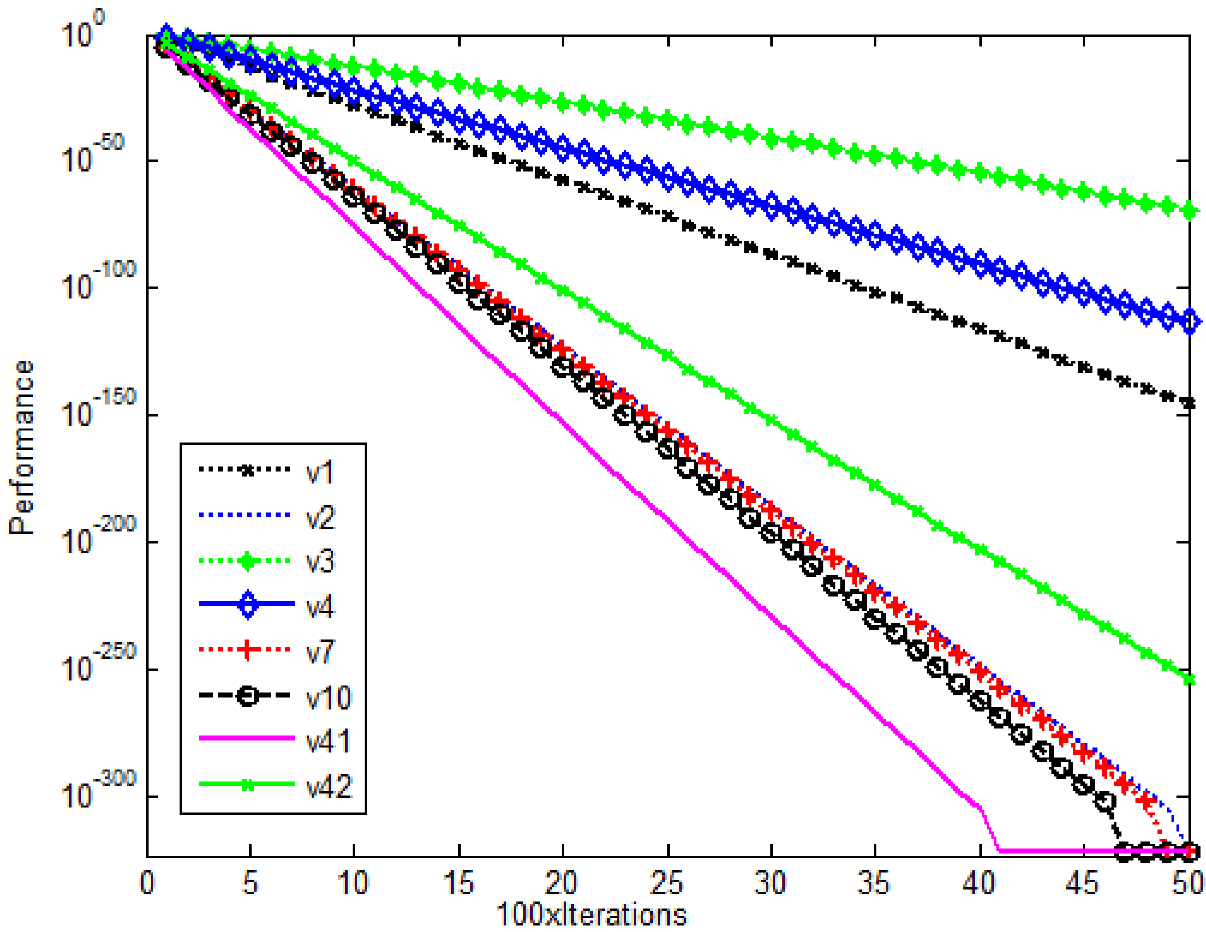


Figure 5.14- 10D average fitness logarithmic convergence graphs for f_{19} showing number of iterations horizontally and average fitness vertically

The average fitness value of *"TSDE/bin" (V41)* and other conventional techniques is presented in Figure 5.14 for Function-15 (f_{19}) optimization problem. This figure shows that *"DE/best/2/bin"(V_4)*, *"DE/rand repeated to best/1/bin"(V_{10})*, *"DE/current to best/1/bin"(V_7)* & *"DE/best/1/bin"(V_2)* reaches to 0 within specified iterations while *"DE/rand/1/bin"(V_1)*, *"DE/rand/2/bin"(V_3)*, *"DE/best/2/bin"(V_4)*, *"TSDE/Exp" (V_{42})* cannot reach at 0 in all 5000 iterations. Mutation strategy *"TSDE/bin" (V41)* reaches to 0 in 4010 iterations while *"DE/rand repeated to best/1/bin"(V_{10})*, *"DE/current to best/1/bin"(V_7)* & *"DE/best/1/bin"(V_2)* is reached at 0 in 4700, 4880 and 4960 iterations respectively. The mutation strategy *"DE/rand/2/bin"(V_3)* has slowest convergence among all strategies. It is important to note that *"TSDE/bin" (V41)* has quick convergence than other strategies illustrated in this figure.

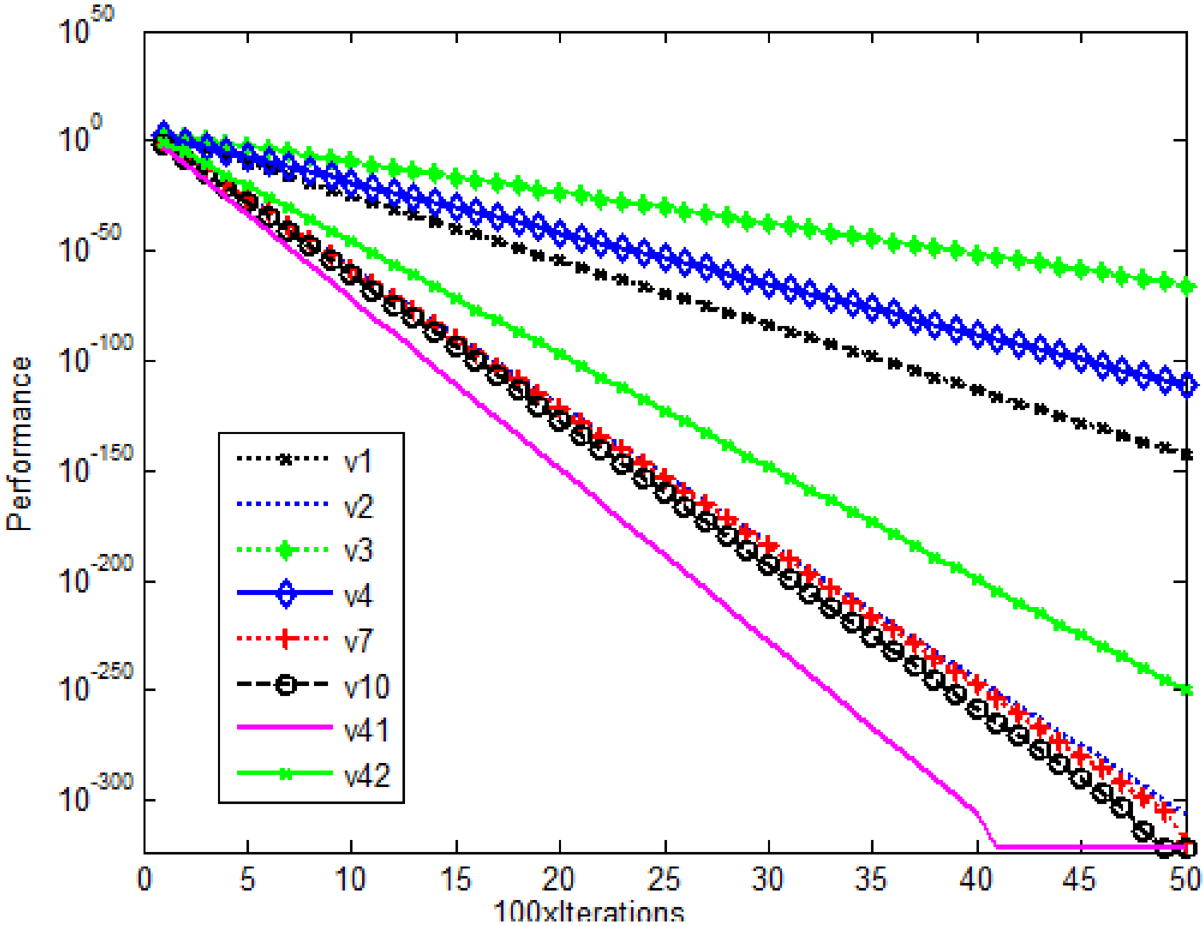


Figure 5.15- 10D average fitness logarithmic convergence graphs for f_{20} showing number of iterations horizontally and average fitness vertically

Figure 5.15 illustrates the average fitness performance of *"TSDE/bin" (V41)* and other mutation strategies for Ellipse-Function (f_{20}). This figure depicts that *"DE/rand/2/bin"(V_3)* has worst performance among all mutation strategies. It is observed from this figure that *"DE/best/1/bin"(V_2)*, *"DE/current to best/1/bin"(V_7)* and *"DE/rand repeated to best/1/bin"(V_{10})* has similar performance in starting iterations however V_{10} convergence is better in succeeding iterations. Mutation strategies *"TSDE/bin" (V41)*, *"DE/rand repeated to best/1/bin"(V_{10})* and *"DE/current to best/1/bin"(V_7)* reach at 0 in 4040, 4750, 4920 iterations respectively. It is important to mention that *"TSDE/bin" (V41)* has quick convergence among all strategies and *"DE/rand repeated to best/1/bin"(V_{10})* has second best performance.

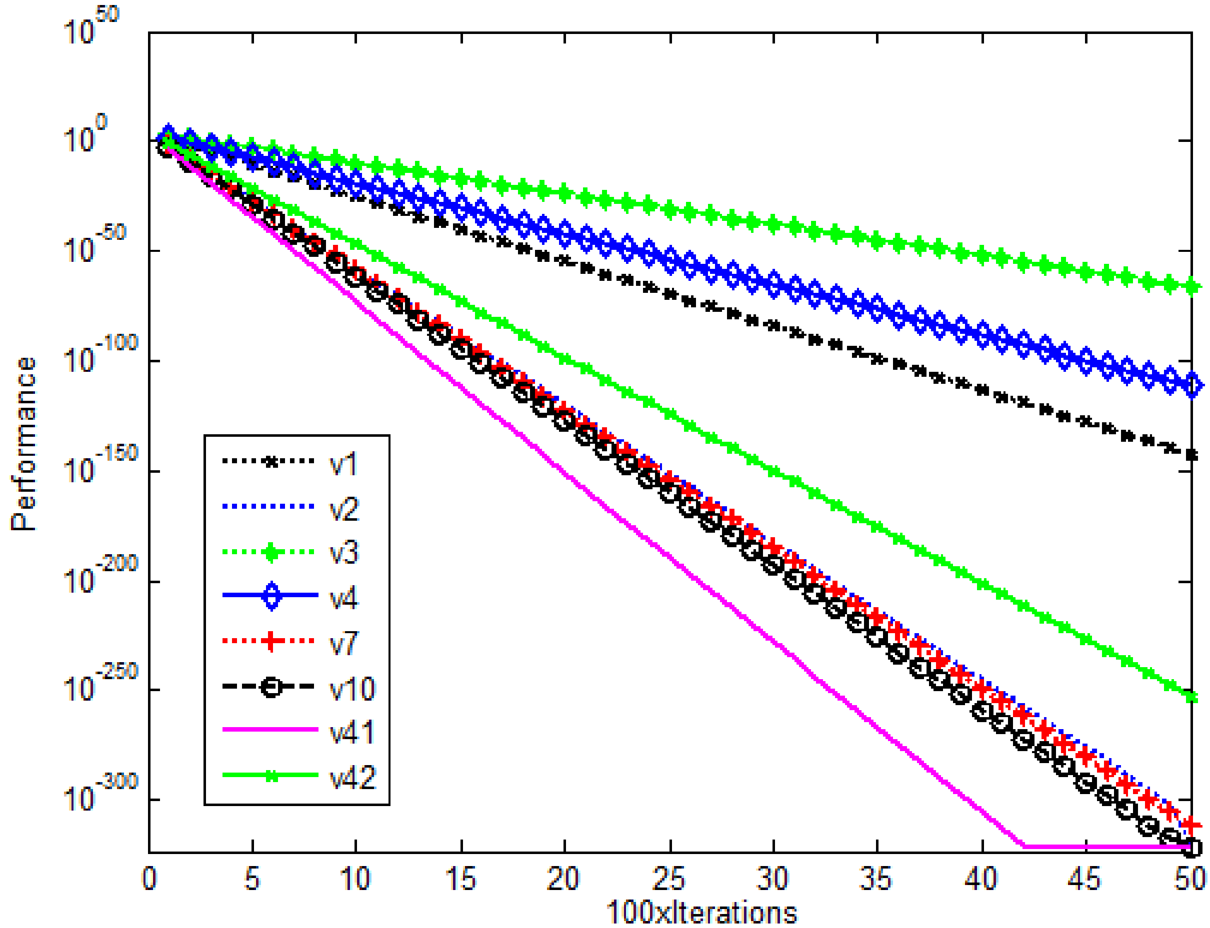


Figure 5.16- 10D average fitness logarithmic convergence graphs for f_{21} showing number of iterations horizontally and average fitness vertically

Average fitness performance of *"TSDE/bin" (V41)* & other mutations strategies is graphically presented in Figure 5.16 for Tablet Function (f_{21}). It is obvious from this figure that *"DE/rand/2/bin"(V_3)* has the worst performance among all mutation strategies. The mutation strategies *"TSDE/bin" (V41)* and *"DE/rand repeated to best/1/bin"(V_{10})* reaches at value 0 within specified iterations. It is important to note that *"TSDE/bin" (V41)* possess best among all mutation strategies as it achieves value 0 in 4040 iterations while *"DE/rand repeated to best/1/bin"(V_{10})* achieve value 0 in 4760 iterations respectively.

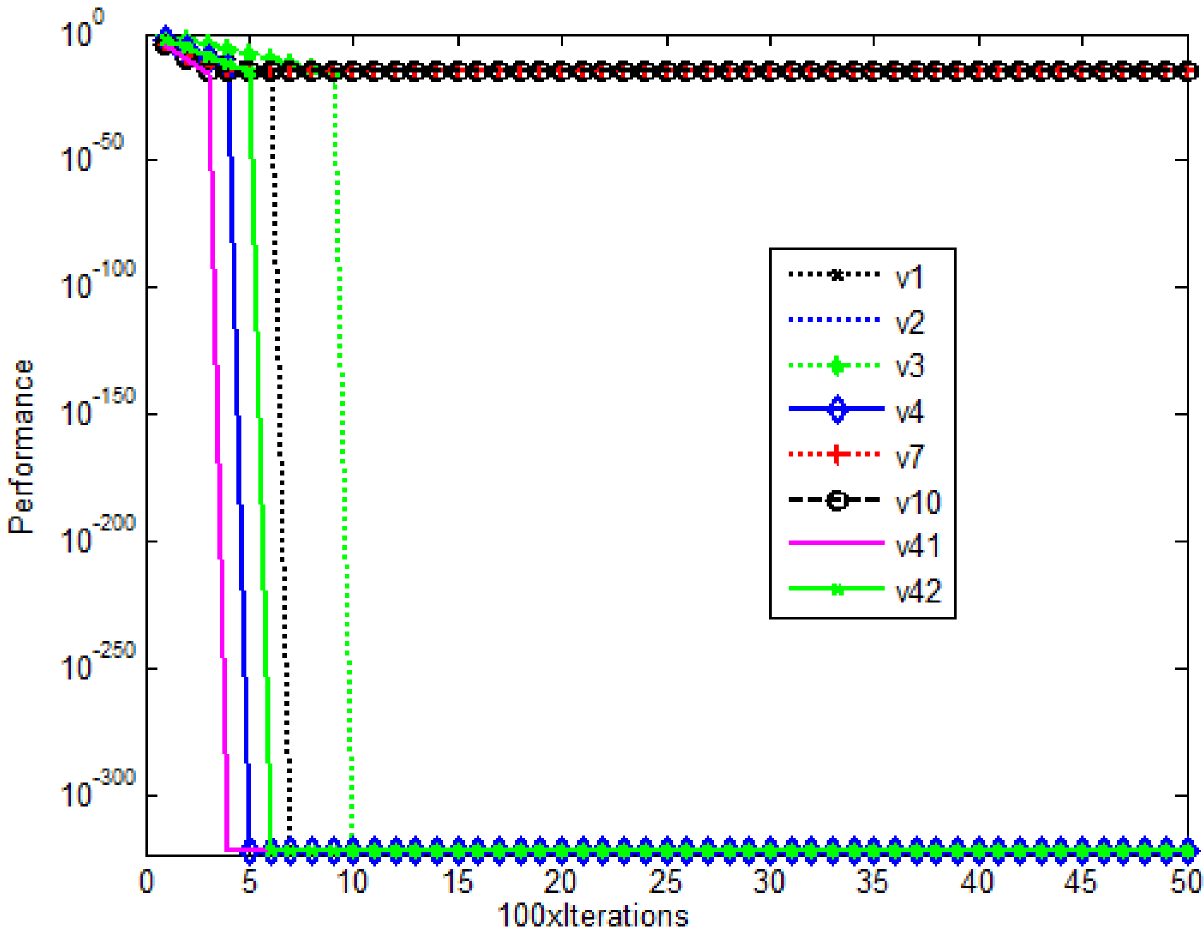


Figure 5.17- 10D average fitness logarithmic convergence graphs for f_{24} showing number of iterations horizontally and average fitness vertically

Average fitness performance of *"TSDE/bin" (V41)* and other mutation strategies is graphically presented in Figure 5.17 for Mishra-1 global optimization problem (f_{24}). This figure shows that *"DE/rand repeated to best/1/bin"(V_{10}), "DE/current to best/1/bin"(V_7)* and *"DE/best/1/bin"(V_2)* have the worst performance among all mutation strategies that reach at values 7.40×10^{-16}, 2.96×10^{-16} and 4.44×10^{-16} in specified 5000 iterations. Mutation strategies *"TSDE/bin" (V41), "DE/best/2/bin"(V_4), "TSDE/Exp" (V_{42}), "DE/rand/1/bin"(V_1)* and *"DE/rand/2/bin"(V_3)* reach at 0 in 380, 490, 610, 690 and 980 iterations respectively. It is important to mention that mutation strategy *"TSDE/bin" (V41)* acquires optimum value in less number of iterations.

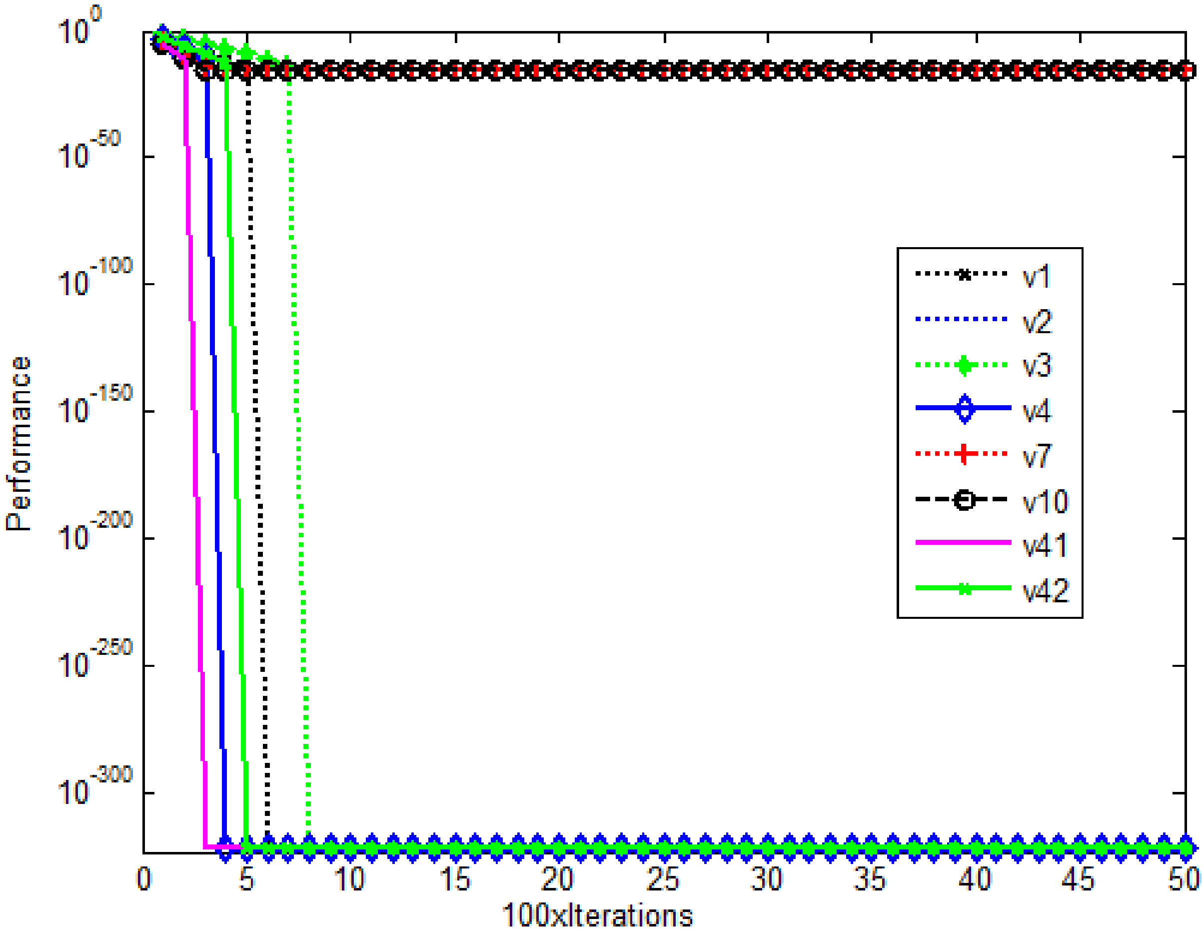


Figure 5.18- 10D average fitness logarithmic convergence graphs for f_{25} showing number of iterations horizontally and average fitness vertically

Average fitness performance of *"TSDE/bin" (V41)* and other mutation strategies is graphically illustrated in Figure 5.18 for Mishra-2 global optimization problem (f_{25}). It is obvious from this figure that *"DE/best/1/bin"(V_2), "DE/current to best/1/bin"(V_7)* and *"DE/rand repeated to best/1/bin"(V_{10})* holds worst performance among all mutation strategies that achieve values 2.22×10^{-15}, 1.04×10^{-15} and 1.63×10^{-15} in the specified iterations. It is evident from this figure that *"TSDE/bin" (V41), "DE/best/2/bin"(V_4), "TSDE/Exp" (V_{42}), "DE/rand/1/bin"(V_1)* and *"DE/rand/2/bin"(V_3)* arrive at optimal value in 300, 460, 540, 580 and 810 iterations respectively. The convergence speed of *"TSDE/bin" (V41)* is quicker than all other strategies illustrated in this figure as it achieves optimal values in less number of iterations.

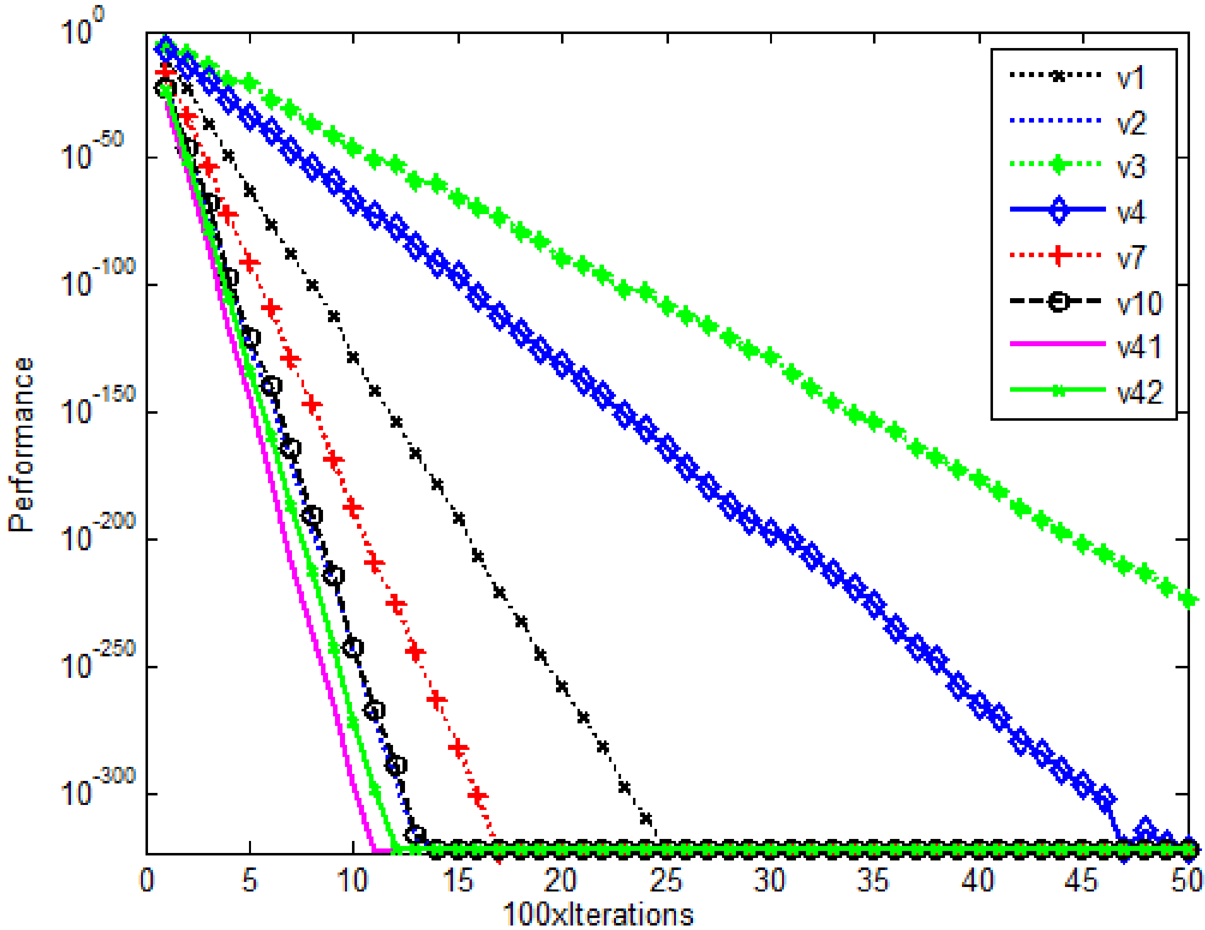


Figure 5.19- 10D average fitness logarithmic convergence graphs for f_{26} showing number of iterations horizontally and average fitness vertically

Figure 5.19 illustrates average fitness performance of *"TSDE/bin" (V41)* & other mutation strategies for MultiModal global optimization problem (f_{26}). It is observed from this figure that *"DE/rand/2/bin"(V_3)* has worst performance among all mutation strategies. Mutation strategies *"TSDE/bin" (V41), "TSDE/Exp" (V_{42}), "DE/and repeated to best/1/bin"(V_{10}), "DE/best/1/bin"(V_2), "DE/current to best/1/bin"(V_7), "DE/rand/1/bin"(V_1)* and *"DE/best/2/bin"(V_4)* arrives at optimum value 0 in 1080, 1190, 1310, 1250, 1650, 2540 and 4790 iterations respectively. It is important to mention that *"TSDE/bin" (V41)* utilizes a less number of iterations to reach at optimum value. Mutation strategy *"TSDE/bin" (V41)* have similar performance to *"TSDE/Exp" (V_{42}), "DE/best/1/bin"(V_2)* and *"DE/rand repeated to best/1/bin"(V_{10})* in early iterations however its decreases more quickly in the subsequent iterations.

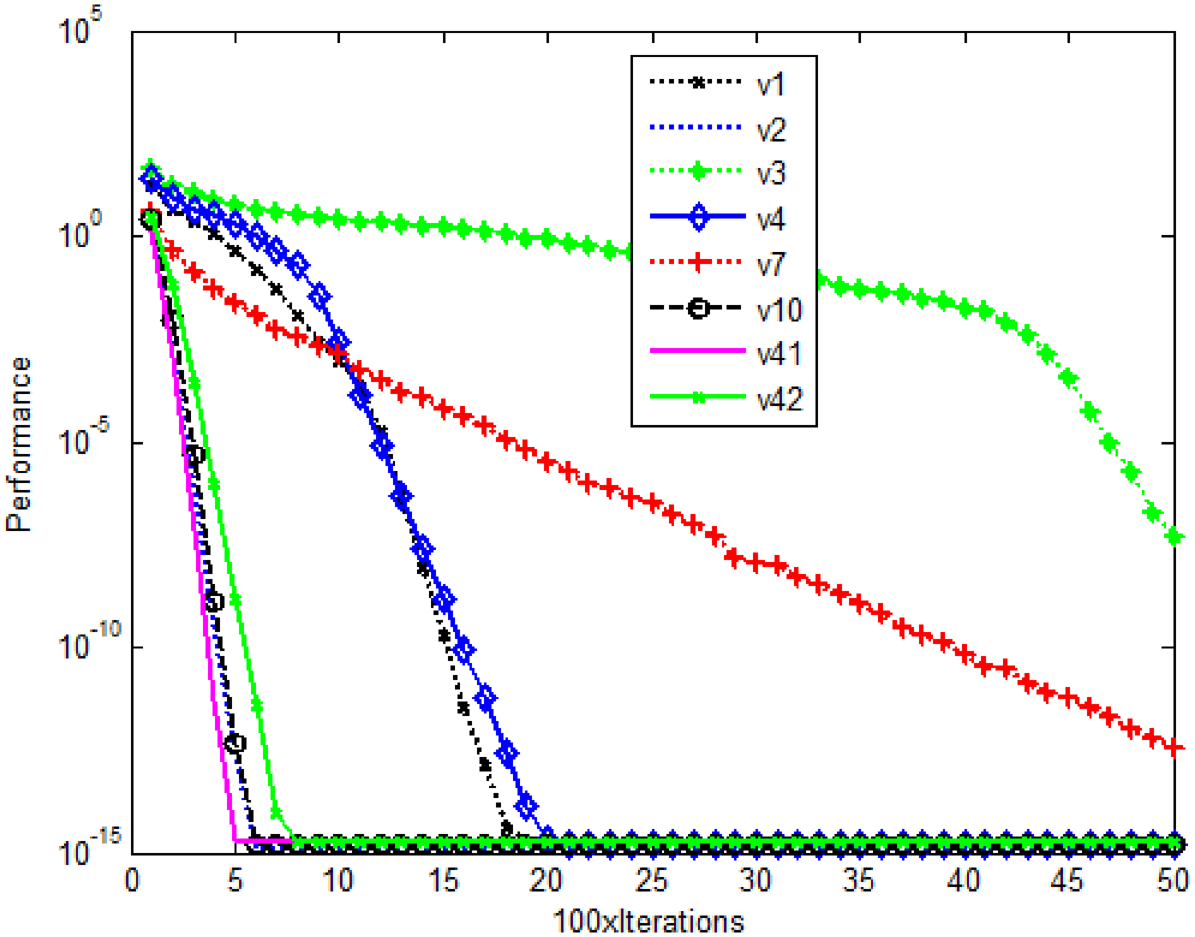


Figure 5.20- 10D average fitness logarithmic convergence graphs for f_{27} showing number of iterations horizontally and average fitness vertically

Average fitness performance of *"TSDE/bin" (V41)* and other mutation strategies is graphically presented in Figure 5.20 for Quintic global optimization problem (f_{27}). It can be observed from this figure that *"DE/rand/2/bin"(V_3)* has the worst performance among all mutation strategies that attains value 4.96×10^{-8} in specified 5000 iterations. The mutation strategy *"TSDE/bin" (V_{41})* has better convergence in early iterations, however its performance is almost similar to *"DE/rand repeated to best/1/bin"(V_{10}), "DE/best/1/bin"(V_2), "TSDE/Exp" (V_{42}), "DE/rand/1/bin"(V_1)* and *"DE/best/2/bin"(V_4)* in the succeeding iterations.

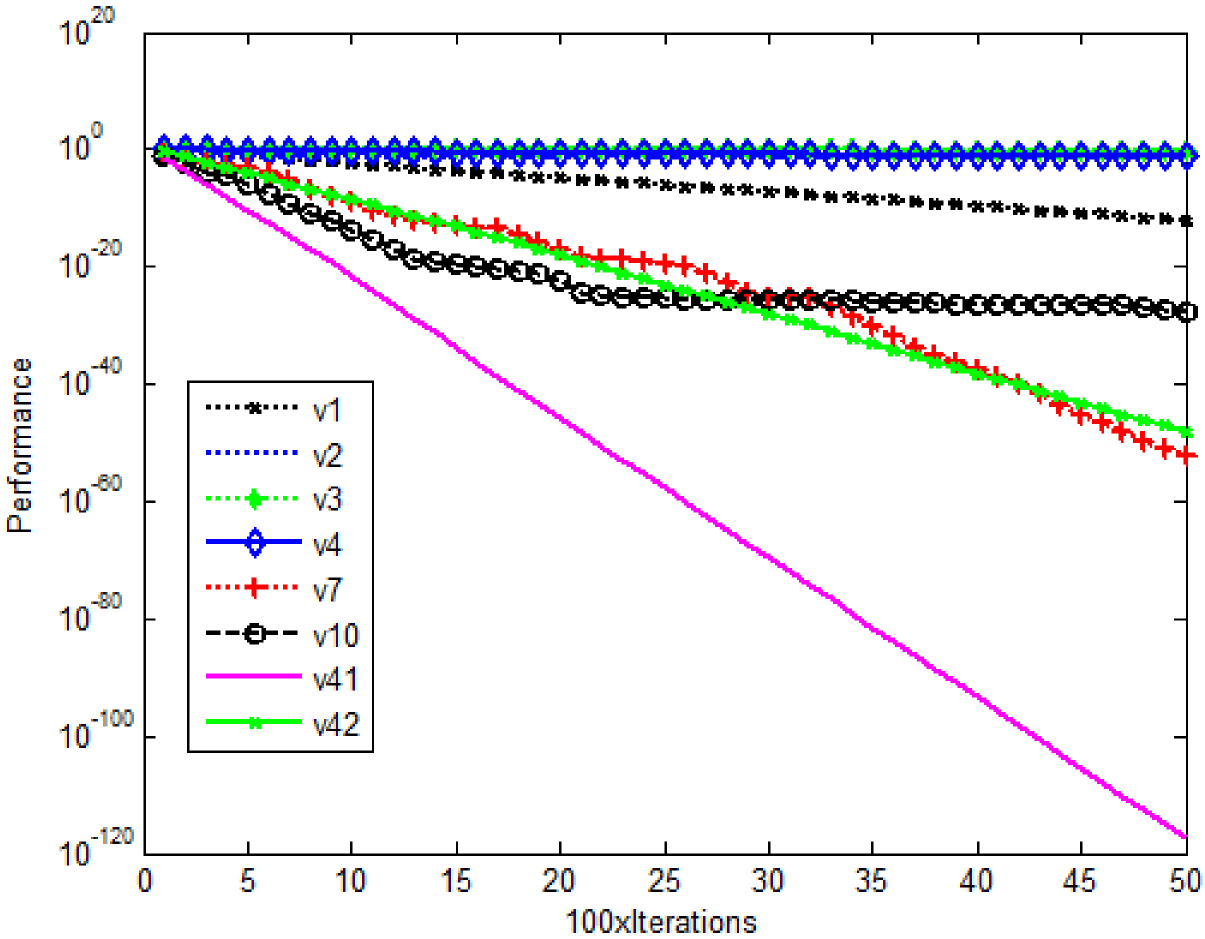


Figure 5.21- 10D average fitness logarithmic convergence graphs for f_{28} showing number of iterations horizontally and average fitness vertically

Figure 5.21 graphically presents an average fitness performance of *"TSDE/bin" (V41)* and other mutation strategies for Stochastic global optimization problem (f_{28}). This figure depicts that *"DE/best/1/bin"(V_2), "DE/rand/2/bin"(V_3)* and *"DE/best/2/bin"(V_4)* have similar and worst performance among all strategies that achieves 2.66×10^{-2}, 8.10×10^{-1} and 8.45×10^{-2} respectively in specified 5000 iterations. It is clear from this figure that the convergence speed of *"TSDE/bin" (V41)* is quicker than all other strategies illustrated in this figure. Mutation strategy *"TSDE/bin" (V41)* continuously decreases from initial iteration to final iteration.

It can be summarized from the simulation results and convergence graphs that the results of TSDE are better than mutation strategies that use the concept of random vectors such as "DE/rand/1", "DE/rand/2" et because these mutation strategies focuses on only diversity that reduces the convergence speed and solution quality of DE algorithm **[**33**]**. By using two best vectors for generating the mutation vectors in DE algorithm enhances the convergence speed and solution quality of TSDE as depicted from the results. The performance of TSDE is better than mutation strategies which uses global best vector such as DE/best/1 and "DE/best/2", "DE/current to best/1", "DE/rand to best/1" since these mutation strategies faces the problem of local optima [66, 85, 118, 169], however, it is proved from the results that inclusion of two best vectors in TSDE make it more powerful to escape from local optima problem and achieve better quality solution. Some mutation strategies uses one difference vectors while TSDE uses two difference vector which makes it capable to produce better perturbation mode than one difference vector mutation strategies **[**120**]**.

5.1.8 Statistical Analysis of the proposed mutation strategy

Dendrogram is an effective way to analyze the similarity/dissimilarity between competitors. The results of DE mutation strategies are analyzed statistically using rank correlation and dendrograms. To find the rank correlation, rank of each DE mutation strategy is calculated to find the correlation between DE strategies. The performance of mutation strategies will be observed during the analysis for various dimensions with the help of dendrogram. The similarity / dissimilarity matrix is used to generate the dendrogram. Similarity matrixes are obtained using "Statistical Package for the Social Sciences" (SPSS). After similarity matrix, pairwise distances between DE strategies are calculated. Pairwise distance is used to create hierarchical cluster tree linkage that is used to create dendrogram. Dendrogram actually generates clusters from the dissimilarity matrix using agglomerative clustering. Dendrogram of DE mutation strategy for a test set of benchmark functions are shown in figures (5.22-29) for different dimensions.

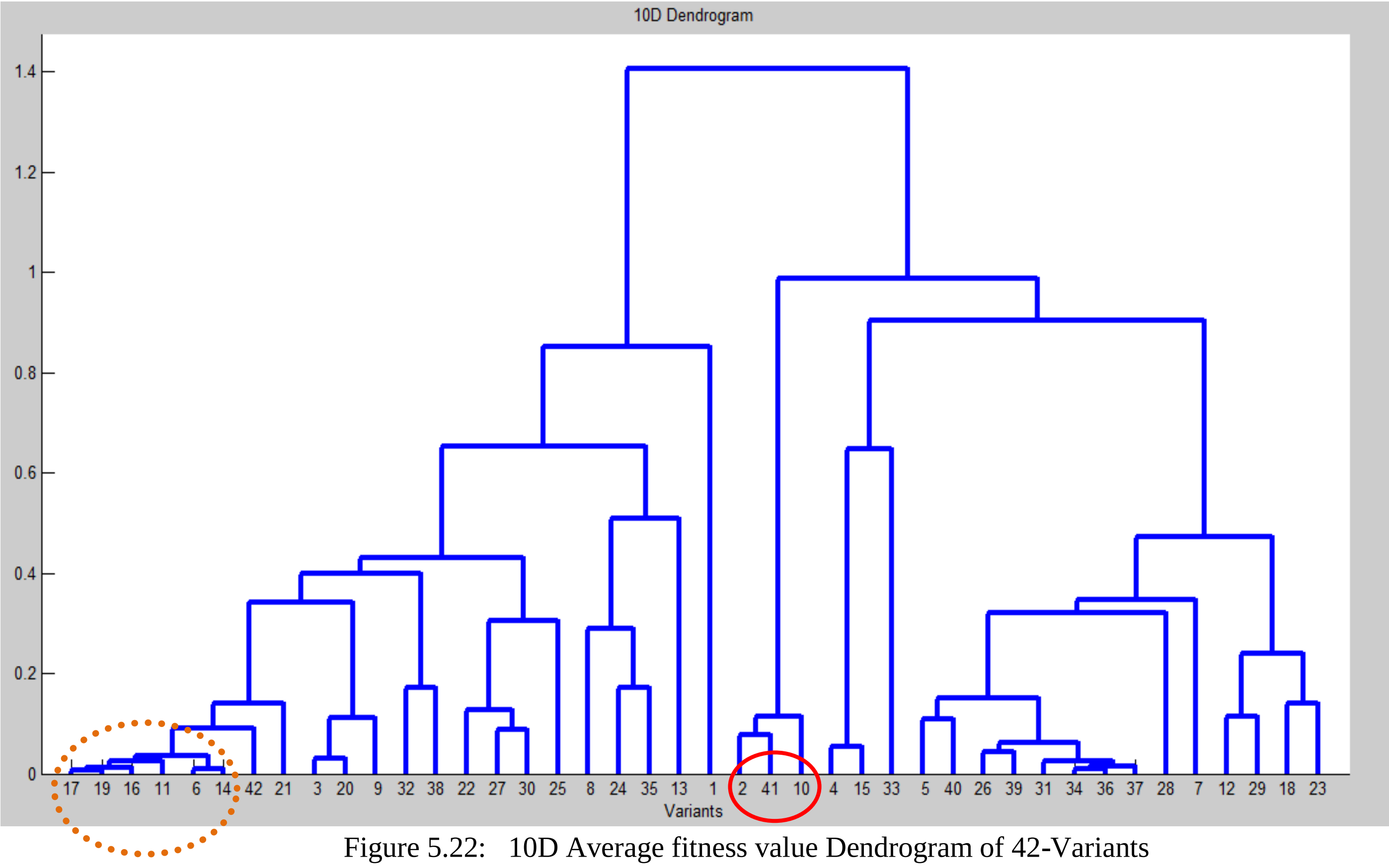


Figure 5.22: 10D Average fitness value Dendrogram of 42-Variants

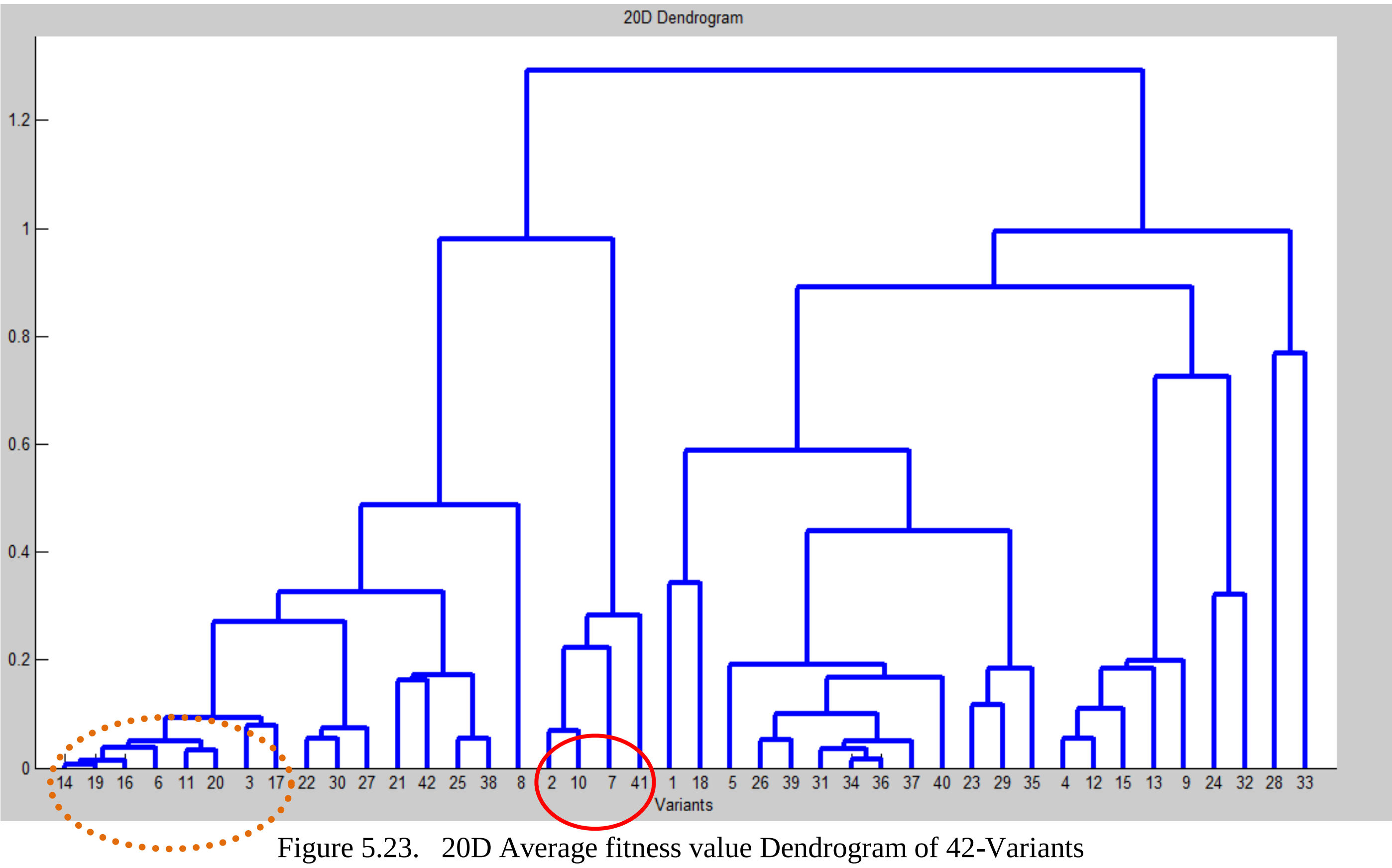


Figure 5.23. 20D Average fitness value Dendrogram of 42-Variants

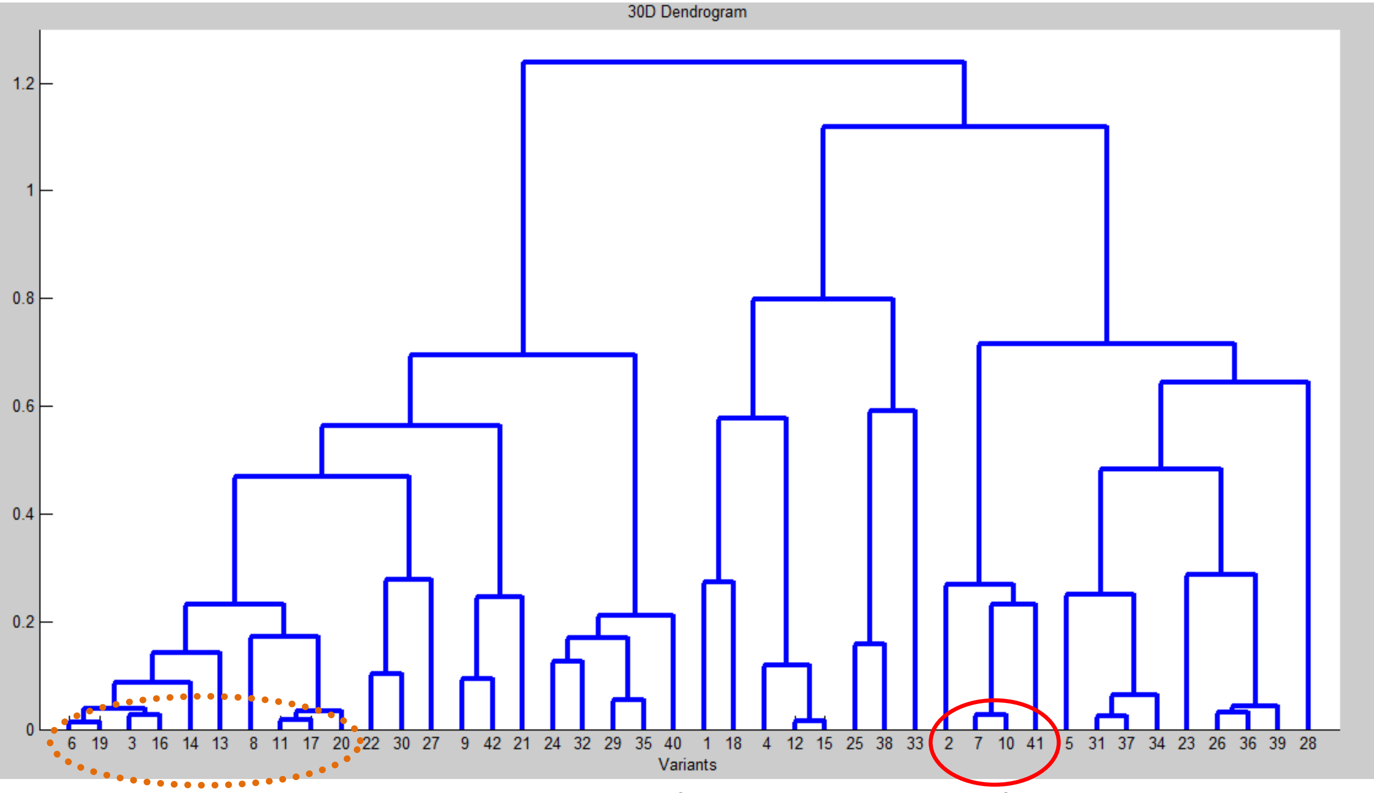


Figure 5.24: 30D Average fitness value Dendrogram of 42-Variants

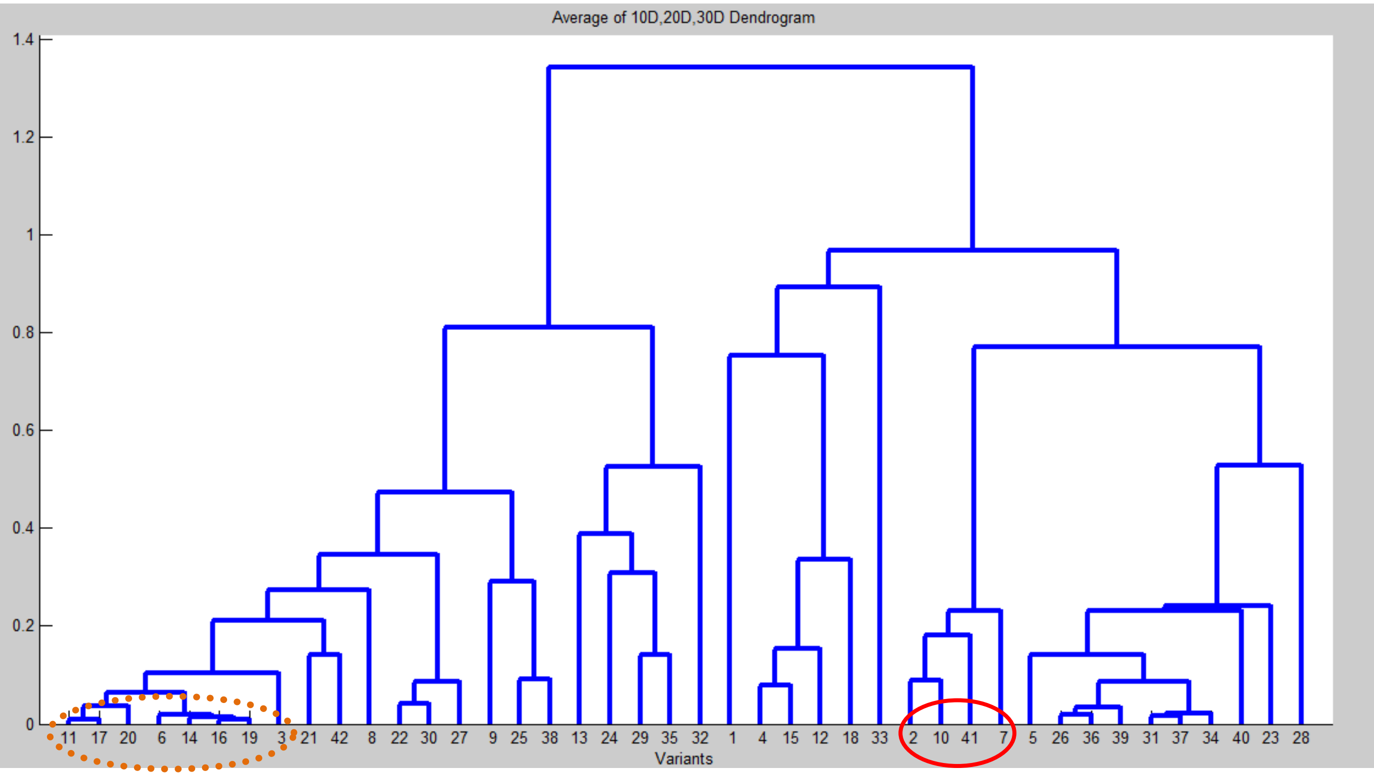


Figure 5.25: Average of average fitness value for 10D, 20D and 30D Dendrogram of 42-DEVariants

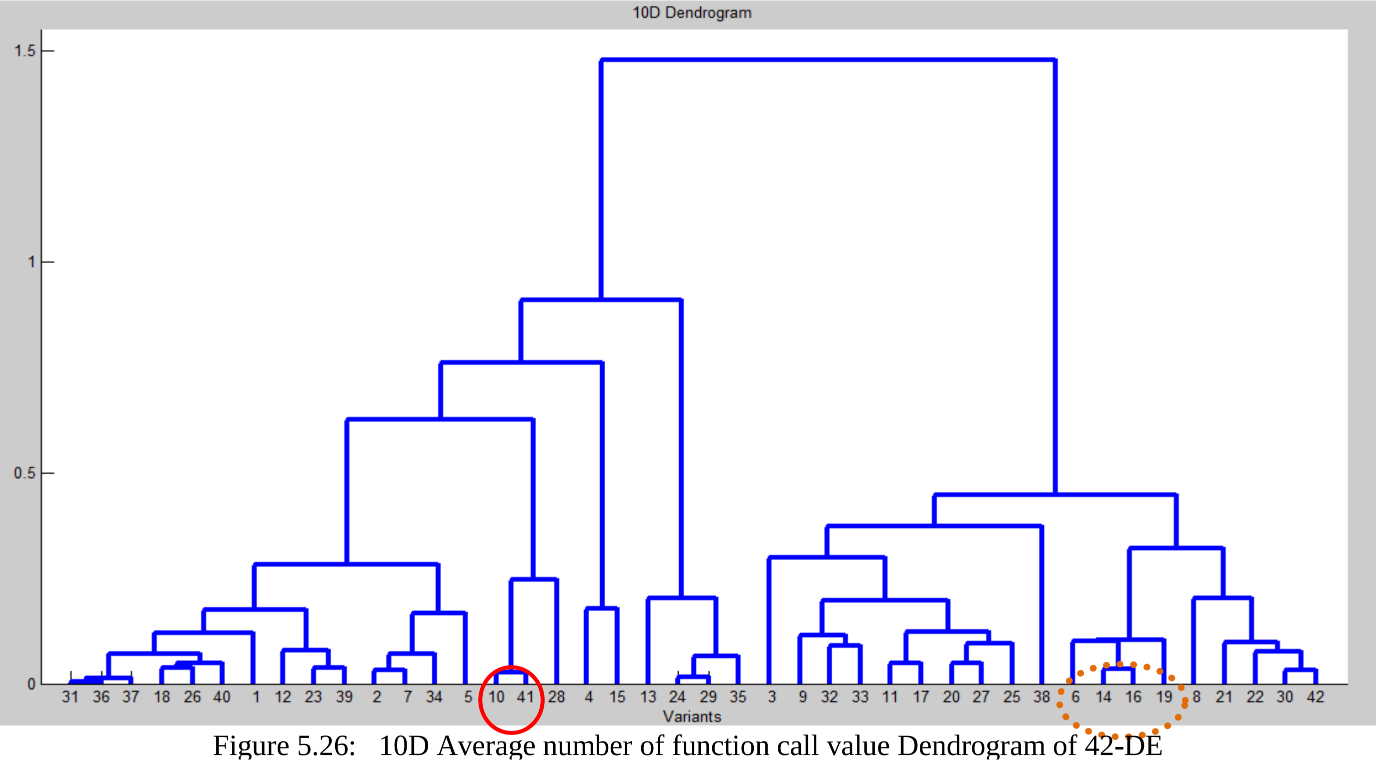


Figure 5.26: 10D Average number of function call value Dendrogram of 42-DE

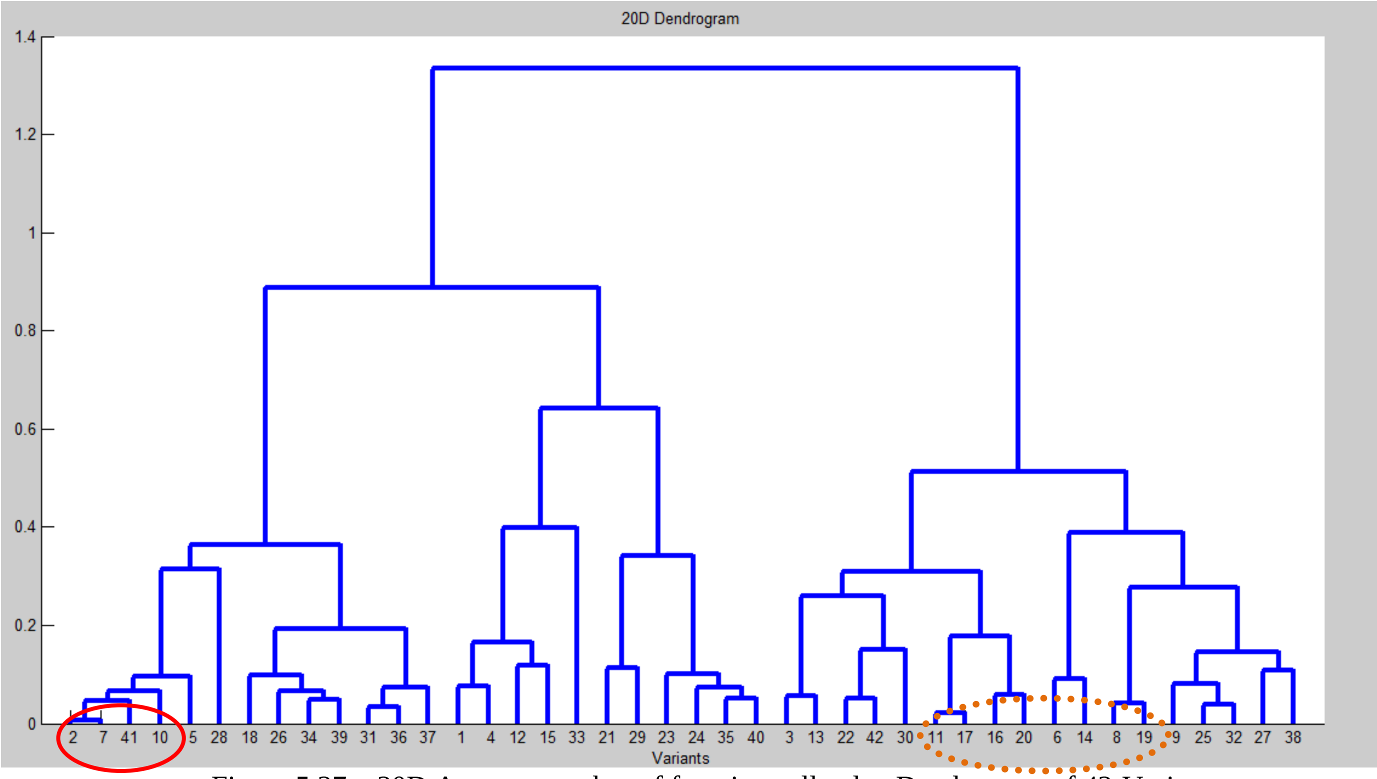


Figure 5.27: 20D Average number of function call value Dendrogram of 42-Variants

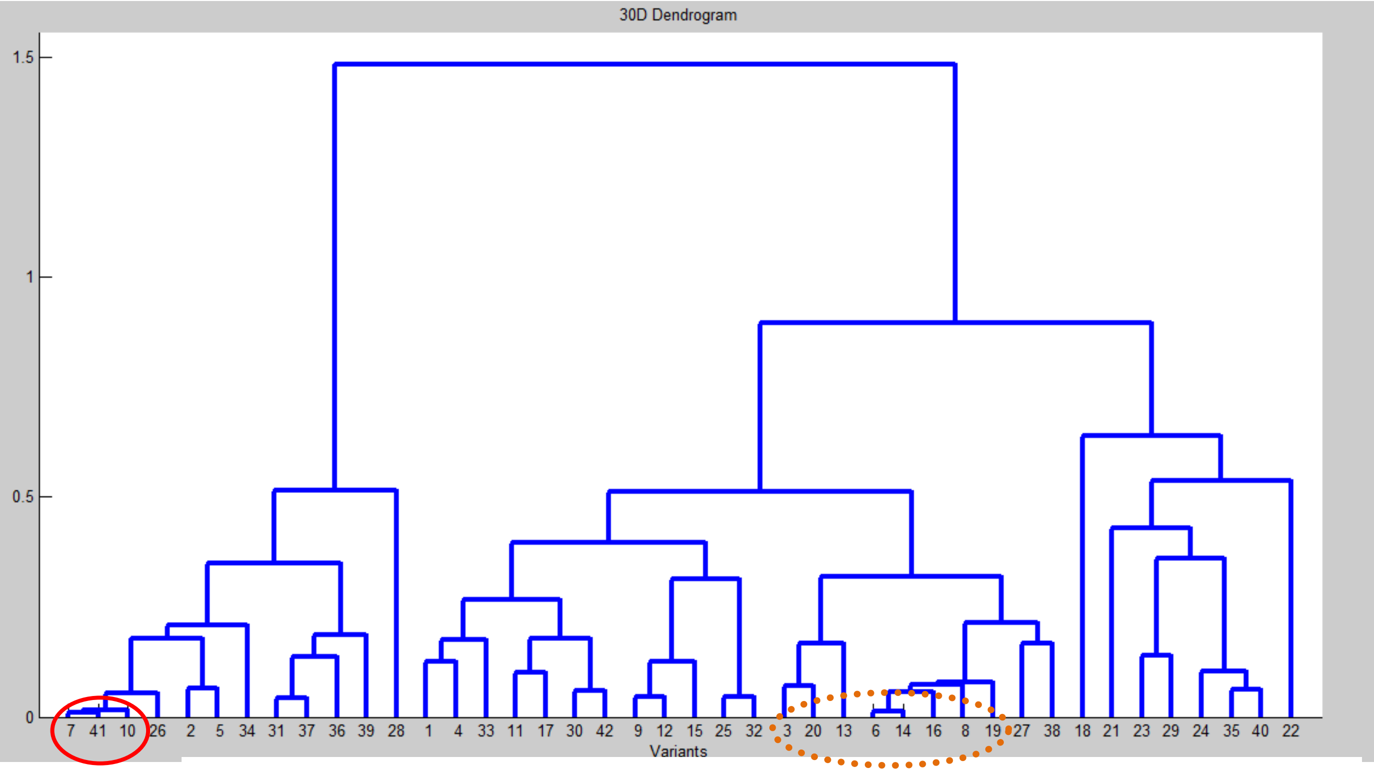


Figure 5.28 30D Average number of function call value Dendrogram of 42-DE Variants

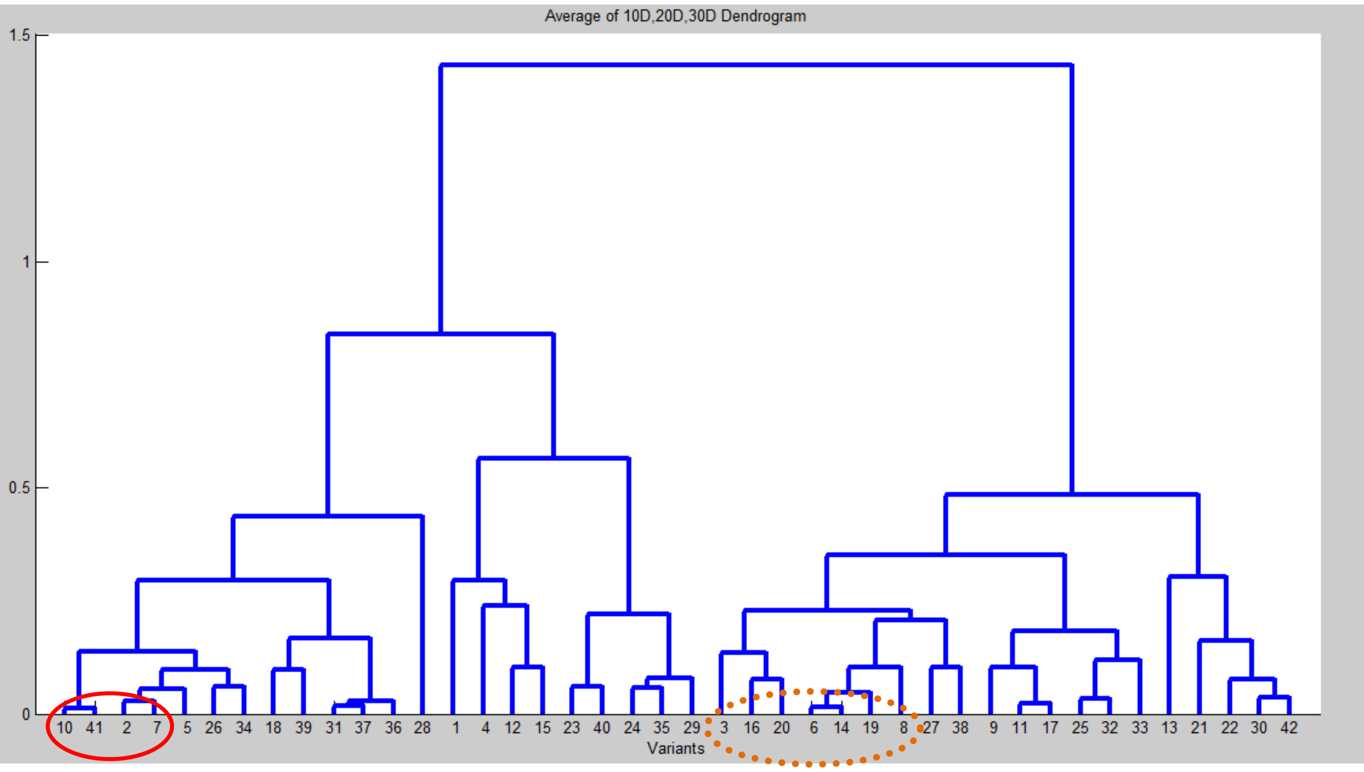


Figure 5.29: Average of average number of function call values for 10D, 20D and 30D Dendrogram of 42-DEVariants

Dendrograms are considered to be an effective way of cluster analysis that specifies the level of similarity joining of any two clusters. The leaf of clusters in Dendrograms contains the DE mutation strategy's number and y-axis contains the dissimilarity scale that shows the joining distance between clusters. The height of each branch in Dendrograms shows that which DE strategies are similar or different from each other. The greater height of DE strategies branches means that strategies are more different from each other. Consider the Dendrograms generated from average fitness values and the number of function call values given in figure-5.22 to figure-5.29 respectively. A similarity measure among the DE strategies is explored by generating Dendrograms from the average fitness values and the number of function call performance parameters. The best performing mutation strategies are contained in the similar performing cluster shown in transparent circles and worst performing strategies are contained in the dotted circle in each Dendrogram. The Dendrograms are generated for 10D, 20D, 30D and the average of all (10D, 20D and 30D) for both performance metrics of fitness value and number of function calls. The best performing as well as worst performing strategies among all mutation strategies are contained in the corresponding groups in most of the cases. The best performing strategy *"TSDE/bin (*V_{41}*)", "DE/rand repeated to best/1/bin (*V_{10}*)"*, *"DE/best/1/bin (*V_2*)"* and *"(*V_7*)"* are contained in the same group in most of the cases that can be considered as best performing mutation strategies that are group shown in the transparent circle in figure-3 to figure-10. The worst performing group of mutation strategies is an encircled dashed line and it contains *"DE/rand repeating & current to rand /1/bin (*V_6*)", "DE/rand & current to best/2/bin(*V_{14}*)", "DE/rand & current to rand /1/bin (*V_{16}*)", "DE/rand/3/bin (*V_{19}*)", "DE/best/3 (*V_{20}*)"* and vice versa.

It is important to discuss that a number of DE mutation strategies are considered in this research work and all these strategies performs different from each other. Dendrograms are also considered to be an effective tool to determine the cluster number. For DE strategies the more clusters in Dendrograms indicates that DE mutation strategies have varying performance from one-another that shows the diverse nature of these strategies in DE algorithm. The diverse nature of DE variants will be helpful to make this algorithm broader applicable in various optimization problems of distinct characteristics.

5.1.9 Comparison of Proposed TSDE with PSO and GA

PSO and GA are two well known heuristics that are applied to a variety of real world optimization problems **[**179**]**. The proposed DE variant *"TSDE/bin (V_{41})"* is compared with PSO and GA through simulation. In order to compare *"TSDE/bin (V_{41})"* with PSO and GA, the following parameter settings are used

GA: Crossover rate CR=0.5 & mutation rate F=0.05 **[**180**]**.

PSO: Inertia weight $\omega \in (0.4, 0.7)$, C1=1.49618, C2=1.49618 are used **[**181**]**.

Table.5.3 Fitness Results (Mean ± S.D) of "TSDE/bin(V_{41})", PSO, GA for functions (f_1- f_{30})

Fun	10D			20D			30D		
	TSDE (V_{41})	**PSO**	**GA**	**TSDE(V_{41})**	**PSO**	**GA**	**TSDE(V_{41})**	**PSO**	**GA**
f_1	**0.00E+00±0.0E+00**	**0.00E+00±0.00E+00**	3.95E-09±1.90E-08	**0.00E+00±0.00E+00**	6.63E-176±0.00E+00	1.17E-04±2.77E-04	**0.00E+00±0.00E+00**	5.24E-01±3.67E+00	2.48E-05±9.55E-05
f_2	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	3.55E-06±8.35E-06	**0.00E+00±0.00E+00**	4.19E+00±1.32E+01	1.05E-03±2.25E-03	**0.00E+00±0.00E+00**	3.15E+01±6.14E+01	1.98E-04±6.17E-04
f_3	9.92E-120±4.77E-118	**1.01E-186±0.00E+00**	1.99E-01±3.35E-01	**1.36E-47±6.90E-47**	4.23E+01±2.96E+02	2.23E+00±2.56E+00	**4.00E-26±6.90E-26**	2.96E+02±8.46E+02	7.18E+00±7.83E+00
f_4	2.66E-01±7.16E-01	2.86E+01±6.47E+01	7.25E+00±1.43E+01	**1.33E-01±7.16E-01**	8.18E+05±7.48E+05	3.82E+01±6.52E+01	**3.99E-01±1.20E+00**	4.13E+06±3.62E+06	3.16E+01±5.14E+01
f_5	1.53E+00±1.17E+00	5.61E+00±6.35E+00	3.98E-08±2.62E-07	6.33E+00±2.03E+00	2.98E+01±2.81E+01	**4.17E-05±8.59E-05**	1.34E+01±3.64E+00	6.09E+01±5.48E+01	**6.11E-06±3.31E-05**
f_6	1.44E-02±1.53E-02	4.54E-02±4.41E-02	4.40E-02±5.85E-02	**2.96E-03±4.82E-03**	2.46E-02±3.46E-02	1.41E-01±1.57E-01	**4.93E-04±1.84E-03**	3.71E+00±1.78E+01	1.64E-01±2.07E-01
f_7	0.00E+00±0.00E+00	1.32E-295±0.00E+00	4.40E-06±6.35E-06	**0.00E+00±0.00E+00**	6.35E-82±4.45E-81	2.29E-06±2.95E-06	**0.00E+00±0.00E+00**	6.29E-61±4.40E-60	8.24E-07±1.23E-06
f_8	9.79E-02±1.01E-01	3.36E-01±3.30E-01	**2.44E-07±4.00E-07**	2.32E-01±6.28E-02	6.48E-01±5.75E-01	**1.98E-05±4.47E-05**	2.73E-01±5.21E-02	1.06E+00±9.64E-01	**4.34E-06±1.43E-05**
f_9	3.31E-07±3.83E-07	**6.09E-18±2.23E-17**	2.50E-04±5.86E-04	**1.55E-07±1.44E-07**	1.27E-02±3.84E-02	5.14E-02±7.54E-02	**1.86E-03±1.00E-02**	6.33E-01±1.99E+00	2.62E-01±2.85E-01
f_{10}	2.11E-151±5.43E-151	**2.82E-220±0.00E+00**	6.17E-03±1.64E-02	3.83E-71±6.60E-71	**2.10E-94±1.43E-93**	2.96E-01±4.39E-01	**1.11E-44±4.40E-44**	4.69E-30±3.13E-29	2.78E-01±3.35E-01
f_{11}	**7.33E-202±0.00E+00**	1.73E-60±1.21E-59	5.54E-30±3.36E-29	**1.67E-232±0.00E+00**	4.00E-01±1.96E+00	1.33E-03±4.88E-03	**1.60E-257±0.00E+00**	9.05E-01±2.99E+00	8.61E-06±5.45E-05
f_{12}	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	6.00E-02±2.37E-01	**0.00E+00±0.00E+00**	3.14E+00±1.01E+01	9.60E-01±1.51E+00	**0.00E+00±0.00E+00**	3.21E+01±7.21E+01	4.46E+00±4.50E+00
f_{13}	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	2.39E-12±7.46E-12	**0.00E+00±0.00E+00**	5.37E-02±3.76E-01	5.08E-09±1.60E-08	**0.00E+00±0.00E+00**	5.37E-01±2.01E+00	2.61E-10±1.00E-09
f_{14}	3.39E-05±5.08E-05	**6.74E-16±1.36E-15**	4.78E-05±1.03E-04	3.05E-05±3.73E-05	**4.74E-15±2.00E-14**	5.82E-05±1.13E-04	1.79E-05±1.77E-05	**2.88E-12±2.02E-11**	3.60E-05±1.26E-04
f_{15}	**3.27E-31±1.31E-46**	1.24E-02±6.09E-02	5.33E-03±6.89E-03	**1.63E-31±6.57E-47**	6.88E-02±2.70E-01	1.23E-02±1.24E-02	**3.46E-03±1.86E-02**	1.88E-01±4.06E-01	4.24E-02±5.34E-02
f_{16}	8.53E+01±7.11E-14	**4.48E+01±2.24E+02**	5.55E+01±4.70E+01	**6.73E-05±7.35E-05**	2.19E+01±1.15E+02	5.26E+01±6.25E+01	**7.43E-05±7.72E-05**	3.94E+05±3.32E+05	2.32E+02±3.02E+02
f_{17}	**0.00E+00±0.00E+00**	2.36E-02±6.17E-02	5.72E-12±1.49E-11	**0.00E+00±0.00E+00**	2.42E-01±2.57E-01	2.08E-06±9.39E-06	1.97E-02±5.02E-02	6.59E-01±6.66E-01	**1.67E-07±5.35E-07**
f_{18}	**0.00E+00±0.00E+00**	2.63E+05±2.37E+05	8.44E-03±4.42E-02	**0.00E+00±0.00E+00**	2.62E+06±2.24E+06	1.99E+01±4.60E+01	**0.00E+00±0.00E+00**	5.62E+06±4.63E+06	1.78E+00±4.46E+00
f_{19}	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	1.84E-08±1.10E-07	**0.00E+00±0.00E+00**	0.00E+00±0.00E+00	4.70E-05±9.73E-05	**0.00E+00±0.00E+00**	7.27E-01±3.56E+00	2.36E-05±1.53E-04
f_{20}	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	9.51E-06±4.62E-05	**0.00E+00±0.00E+00**	3.74E-202±0.00E+00	1.96E-02±4.10E-02	**0.00E+00±0.00E+00**	9.98E-05±6.86E-04	7.45E-03±3.35E-02
f_{21}	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	1.20E-03±5.65E-03	**0.00E+00±0.00E+00**	3.85E-134±2.70E-133	1.24E-01±2.93E-01	**0.00E+00±0.00E+00**	7.66E-09±3.70E-08	1.03E-01±6.10E-01
f_{22}	**0.00E+00±3.38E-32**	**0.00E+00±0.00E+00**	5.05E-01±5.19E-01	**1.51E-31±1.56E-31**	1.19E-07±7.97E-07	2.93E+00±2.84E+00	**6.86E-31±4.01E-31**	4.79E+04±5.51E+04	5.45E+00±4.63E+00
f_{23}	**1.56E-06±1.32E-06**	1.92E+00±1.66E+00	7.67E-03±1.90E-02	**1.57E-06±1.36E-06**	5.64E+00±4.69E+00	6.95E-02±1.17E-01	**1.36E-06±1.30E-06**	9.25E+00±7.67E+00	1.71E+00±1.55E+00
f_{24}	**0.00E+00±0.00E+00**	0.00E+00±0.00E+00	7.15E-01±1.64E+00	**0.00E+00±0.00E+00**	0.00E+00±0.00E+00	1.21E+02±4.28E+02	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	2.95E+04±1.43E+05
f_{25}	**0.00E+00±0.00E+00**	0.00E+00±0.00E+00	1.16E+00±1.68E+00	**0.00E+00±0.00E+00**	0.00E+00±0.00E+00	5.86E+01±1.54E+02	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	5.10E+04±1.61E+05
f_{26}	**0.00E+00±0.00E+00**	0.00E+00±0.00E+00	0.00E+00±0.00E+00	**0.00E+00±0.00E+00**	0.00E+00±0.00E+00	0.00E+00±0.00E+00	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	0.00E+00±0.00E+00
f_{27}	1.81E-15±1.98E-16	**6.31E-16±5.72E-16**	1.63E+00±1.48E+00	**3.42E-15±3.81E-16**	2.07E-01±1.39E+00	5.35E+00±4.52E+00	**5.13E-15±2.20E-15**	1.62E+00±3.02E+00	1.50E+01±1.25E+01
f_{28}	**7.35E-118±1.46E-117**	8.31E-02±1.41E-01	1.42E-66±9.95E-66	**4.98E-118±1.94E-117**	9.70E-01±1.04E+00	4.70E-19±3.29E-18	1.74E-120±7.93E-120	3.43E+00±3.27E+00	**4.27E-235±0.00E+00**
f_{29}	9.43E-08±7.09E-07	**1.87E-29±3.05E-29**	8.32E-10±2.89E-09	1.11E-07±4.98E-07	**1.65E-29±2.73E-29**	3.25E-09±9.17E-09	2.04E-07±9.92E-07	**2.94E-29±4.86E-29**	2.43E-10±6.39E-10
f_{30}	**2.10E-04±4.68E-06**	4.02E-04±3.78E-04	1.68E-03±3.62E-03	**2.64E-08±4.35E-09**	5.36E-08±4.59E-08	1.13E-04±2.58E-04	**2.54E-12±6.83E-13**	4.83E-12±4.22E-12	2.39E-05±7.86E-05

Table.5.4 Number of Function Call's (Mean ± S.D) of TSDE/bin(V_{41}), PSO, GA for functions (f_1- f_{30})

Fun	10D			20D			30D		
	TSDE(V_{41})	PSO	GA	TSDE(V_{41})	PSO	GA	TSDE(V_{41})	PSO	GA
f_1	**8.38E+01±4.86E+00**	8.70E+01±6.98E+00	1.94E+03±7.64E+02	**1.56E+02±5.21E+00**	1.78E+02±2.56E+01	1.22E+04±3.35E+03	**2.21E+02±7.02E+00**	3.74E+02±9.70E+01	1.23E+04±2.88E+03
f_2	**8.47E+01±3.34E+00**	8.86E+01±6.04E+00	4.47E+03±2.31E+03	**1.69E+02±5.48E+00**	2.00E+02±3.74E+01	2.54E+04±8.90E+03	**2.46E+02±7.61E+00**	4.81E+02±1.64E+02	1.81E+04±5.80E+03
f_3	3.30E+02±1.91E+01	**2.18E+02±2.01E+01**	5.64E+04±1.68E+04	1.72E+03±1.05E+02	**9.45E+02±1.15E+02**	-	4.57E+03±2.43E+02	**2.80E+03±3.27E+02**	2.11E+05±3.81E+04
f_4	**1.23E+03±5.85E+01**	3.39E+04±1.33E+04	-	**2.84E+03±4.03E+02**	-	-	**4.78E+03±5.89E+02**	-	-
f_5	**5.21E+02±1.85E+02**	-	2.04E+03±6.31E+02	-	-	**9.05E+03±2.46E+03**	-	-	**1.27E+04±4.54E+03**
f_6	**3.92E+02±2.44E+02**	-	1.90E+04±2.19E+04	**2.36E+02±1.67E+01**	2.53E+02±3.29E+01	7.26E+04±4.95E+04	**3.17E+02±1.14E+01**	5.63E+02±4.44E+01	4.48E+04±2.83E+04
f_7	**4.50E+01±3.38E+00**	5.85E+01±7.33E+00	3.59E+03±2.77E+03	**6.14E+01±4.93E+00**	7.21E+01±8.89E+00	3.70E+03±4.44E+03	**7.06E+01±5.21E+00**	8.74E+01±1.23E+01	2.05E+03±3.22E+03
f_8	**5.53E+02±1.32E+02**	-	2.36E+03±7.19E+02	-	-	1.11E+04±3.32E+03	-	-	**1.20E+04±4.23E+03**
f_9	2.40E+02±1.66E+02	**8.11E+01±2.11E+01**	3.97E+04±2.96E+04	2.24E+02±1.17E+02	**1.13E+02±2.21E+01**	1.08E+05±4.20E+04	2.53E+02±1.22E+02	**1.47E+02±0.00E+00**	-
f_{10}	2.68E+02±1.31E+01	**1.77E+02±1.61E+01**	3.24E+04±1.94E+04	1.26E+03±5.44E+01	**8.96E+02±4.40E+02**	-	**2.96E+03±1.28E+02**	2.98E+03±7.84E+02	2.72E+05±3.36E+03
f_{11}	**1.61E+02±4.08E+00**	1.77E+02±1.49E+01	6.93E+02±6.42E+02	**2.96E+02±6.82E+00**	7.51E+02±4.64E+02	1.30E+04±9.92E+03	**4.17E+02±1.15E+01**	-	4.72E+03±7.16E+03
f_{12}	**5.66E+01±5.33E+00**	8.13E+01±9.73E+01	2.52E+03±1.36E+03	**1.07E+02±5.02E+00**	2.44E+04±4.54E+04	2.00E+04±9.05E+03	**1.50E+02±7.71E+00**	1.08E+05±1.40E+05	8.14E+04±4.37E+04
f_{13}	3.69E+01±4.15E+00	**3.50E+01±3.94E+00**	1.28E+02±6.27E+01	**8.27E+01±4.63E+00**	8.68E+01±8.17E+00	7.40E+02±3.59E+02	**1.29E+02±6.28E+00**	1.70E+02±2.21E+01	2.00E+03±9.29E+02
f_{14}	1.80E+04±1.58E+04	**1.34E+02±3.92E+01**	2.12E+04±2.55E+04	1.91E+04±1.98E+04	**1.53E+02±5.99E+01**	4.70E+04±4.92E+04	3.93E+04±2.82E+04	**2.11E+02±2.94E+02**	4.33E+04±6.18E+04
f_{15}	**8.10E+01±5.75E+00**	8.20E+01±7.37E+00	-	**1.44E+02±6.32E+00**	2.05E+02±7.46E+01	-	**1.98E+02±1.01E+01**	7.29E+02±6.37E+02	-
f_{16}	-	**3.83E+02±5.45E+01**	-	4.42E+04±4.67E+04	**2.62E+03±1.89E+03**	-	**1.18E+05±7.21E+04**	-	-
f_{17}	**8.36E+01±3.87E+00**	8.82E+01±8.18E+00	7.28E+02±3.00E+02	1.58E+02±7.40E+00	**1.54E+02±7.07E-01**	7.18E+03±2.89E+03	**2.18E+02±7.89E+00**	-	8.88E+03±3.05E+03
f_{18}	**1.57E+02±4.75E+00**	-	5.59E+03±1.05E+03	**2.80E+02±7.64E+00**	-	2.89E+04±3.68E+03	**3.94E+02±9.44E+00**	-	3.14E+04±5.82E+03
f_{19}	**8.26E+01±4.92E+00**	8.38E+01±5.33E+00	1.87E+03±5.35E+02	**1.54E+02±5.76E+00**	1.76E+02±2.02E+01	1.10E+04±3.49E+03	**2.18E+02±6.55E+00**	4.70E+02±3.24E+02	1.25E+04±2.75E+03
f_{20}	**1.16E+02±5.17E+00**	1.21E+02±5.81E+00	3.38E+03±1.00E+03	**2.12E+02±7.97E+00**	2.49E+02±2.46E+01	2.05E+04±4.16E+03	**3.02E+02±7.37E+00**	6.98E+02±3.67E+02	2.11E+04±4.24E+03
f_{21}	**1.25E+02±4.85E+00**	1.51E+02±8.11E+00	4.88E+03±1.05E+03	**2.20E+02±6.22E+00**	2.74E+02±2.99E+01	2.53E+04±5.45E+03	**3.05E+02±8.38E+00**	8.12E+02±4.91E+02	2.59E+04±6.54E+03
f_{22}	**1.37E+02±6.72E+00**	1.64E+02±1.14E+01	-	**2.98E+02±2.05E+01**	3.01E+03±1.52E+03	-	**5.07E+02±3.02E+01**	2.81E+05±-	-
f_{23}	**1.27E+03±1.25E+03**	-	6.54E+04±1.43E+04	**1.42E+03±1.59E+03**	-	9.50E+04±0.00E+00	**2.34E+03±1.89E+03**	-	-
f_{24}	9.76E+01±4.62E+00	**3.03E+00±3.20E-01**	8.16E+03±3.32E+03	1.83E+02±4.67E+00	**3.27E+00±5.21E-01**	3.26E+04±1.31E+04	2.60E+02±7.35E+00	**3.40E+00±4.98E-01**	6.19E+04±1.94E+04
f_{25}	1.04E+02±3.15E+00	**3.13E+00±3.46E-01**	1.03E+04±4.17E+03	1.89E+02±4.89E+00	**3.37E+00±5.56E-01**	3.21E+04±6.75E+03	2.68E+02±6.37E+00	**3.57E+00±5.04E-01**	7.08E+04±2.16E+04
f_{26}	2.12E+01±4.78E+00	**3.03E+00±1.83E-01**	2.74E+01±9.37E+00	**3.12E+01±1.33E+01**	**3.00E+00±0.00E+00**	3.24E+01±1.60E+01	3.65E+01±5.91E+00	**2.87E+00±3.46E-01**	1.57E+01±6.78E+00
f_{27}	2.31E+02±2.05E+01	**2.18E+02±2.38E+01**	-	**4.09E+02±2.27E+01**	9.27E+02±2.96E+02	-	**5.49E+02±2.28E+01**	-	-
f_{28}	**2.53E+02±1.42E+01**	-	1.31E+03±9.83E+02	**5.12E+02±1.81E+01**	-	1.59E+03±8.26E+02	**7.79E+02±2.87E+01**	-	1.30E+03±8.97E+02
f_{29}	3.42E+01±3.39E+01	**2.23E+01±1.53E+01**	4.80E+01±5.54E+01	3.45E+01±3.05E+01	**2.38E+01±1.36E+01**	2.19E+02±5.88E+02	3.05E+01±2.44E+01	**1.61E+01±1.33E+01**	8.63E+01±1.19E+02
f_{30}	-	-	-	6.68E+01±2.77E+01	**1.16E+01±8.32E+00**	3.41E+04±5.31E+04	1.12E+01±6.17E+00	**3.03E+00±1.03E+00**	2.85E+03±8.15E+03

Same parameters like population size, dimensions, number of iterations and standard benchmark functions are used in the comparison of "TSDE/bin (V_{41})" with PSO, GA. Results of average fitness value and NFC are presented in table-5.3 and table-5.4 respectively. Proposed strategy "TSDE/bin(V_{41})" has dominating performance than evolutionary heuristics PSO and GA in most of the cases of 10D, 20D and 30D for average fitness value and NFC. "TSDE/bin(V_{41})" proves itself to be one of the powerful variant of DE algorithm that can be a significant addition in DE research.

5.1.10 Summary

In this chapter a novel technique TSDE mutation strategy based on the selection of parent vectors to generate the new population is proposed in this research. The proposed variants *“TSDE/bin” (V_{41})* and *“"TSDE/exp" (V_{42})”* are compared with the existing DE mutation strategies. *“TSDE/bin” (V_{41})* mutation variant has dominating performance among the DE mutation strategies. This new mutation variant *TSDE* will prove to be a valuable addition to DE literature. A comprehensive set of well known N-dimensional benchmark functions have been used to evaluate the performance of the proposed *TSDE* as well as existing mutation strategies of DE algorithm. In this research, extensive comparison is performed by considering the most number of mutation strategies; such a comparison is missing in the literature. This comparison unveils some very beneficial facts related to the DE research improvements. The other important aspect of this research is the statistical comparison of the DE mutation strategies; such a comparison is also not available in the literature. Experimental results consist of average fitness value and number of function call parameter of DE algorithm that are reported in tables (5.1-5.2) and their sub-tables. Since mutation strategies are forty two and it is not possible to report all the mutation strategies in few tables, so proposed TSDE mutation variant, five commonly used mutation strategies and one better performing mutation strategy are reported in the results. It can be concluded that there is a deviation in the performance of DE algorithm variants and the selection of DE variant affects the performance result of DE algorithm. Better results can be obtained by choosing the better performing strategies. The proposed binomial (*“TSDE/bin (V_{41})”*) has the leading performance among the DE mutation strategies for both fitness value and number of function call performance parameters, the proposed exponential *“TSDE/exp”(V_{42})*, mutation strategy *“DE/rand to best/1/bin(V_{10})”, “"DE/current to best/1/bin"(V_7)”* and *“DE/best/1/bin (V_2)”* are the other better performance mutation strategies. This research work also reveals one of the best performing mutation strategy *“DE/rand to best/1/bin (V_{10})”* that has been rarely brought into play. This research work will prove to be a significant addition to DE literature. The results of *“TSDE/bin (V_{41})”* are also compared with two well known heuristics GA and PSO that shows dominating performance of *“TSDE/bin (V_{41})”*.

Chapter # 6: Proposed Random controlled Pool base Differential Evolution algorithm (RCPDE)

6.0 Chapter Summary

This chapter presents another proposed random controlled pool based differential evolution algorithm. The proposed algorithm is tested against standard benchmark functions and the results are reported in table form. The proposed random controlled pool based differential evolution algorithm is compared with other state of the art DE algorithms.

6.1 Introduction

DE algorithm has different mutation strategies that behave differently when solving different problems. The mutation strategy pool contains mutation strategies having diverse characteristics. The strategy pool consists of effective trial vector mutation strategies taken from **[**120**]** and previous work of author **[**177**]**. Various researchers widely investigated control parameters and DE mutation strategies in the last decade and some prior knowledge has obtained. The prior knowledge is helpful in designing control parameter pool and strategy pool in of DE algorithm. In proposed RCPDE algorithm we have used five mutation strategies in the strategy candidate pool and four control parameters to form a control parameter candidate pool taken from [120, 177, 182-83] of DE algorithm. The varying behavior of control parameters of parameter pool and strategies of strategy pool will be helpful in solving different kinds of problems.

6.2 Significance of proposed mutation RCPDE

The significance of proposed mutation strategy pool and control parameter pool of DE algorithm is discussed in the following subsections.

6.2.1 Strategy Pool

1. DE/rand to best/1: This mutation strategy utilizes the information of best solutions that is helpful in fast convergence speed **[**161**]**. Moreover, it is helpful because it uses two difference vectors and perturbs random vector in the direction of best vector that may incorporate more diversity by producing more trial vector**s** [33, 120, 161]. DE/rand to best/1 mutation strategy prove itself to be one of the better performing mutation strategy **[**177**]**.
2. DE/rand/1: It bears stronger exploration in the DE algorithm that is helpful to incorporate diversity in the population [33]

3. DE/current to rand/1: It is used to solve the rotated problems more effectively than other strategies [33, 184]
4. DE /rand/2: It is helpful to improve diversity as well, it perturbs random vector and used two difference vectors that are helpful to incorporate more diversity than one difference vector by producing more trial vectors [33, 117, 161]
5. TSDE/bin: It is helpful to improve the convergence speed of DE algorithm and will be helpful to avoid from local optima problem **[**177**]**.

6.2.2 DE Control parameter Pool

Crossover generates new solutions by shuffling competing vectors information between population individuals that increases increase the population diversity and also increase the opportunities to reproduce superior individuals in the current population [185-86]. The smaller values of CR is helpful in solving separable problems while large values of CR is helpful in solving multimodal problems [187-88]. New information in the population is introduced by the mutation operation that generates random variations in the population individuals **[**189**]**. The smaller values of F is helpful for exploitation and Large F is helpful to maintain Exploration in DE algorithm **[**33**]**. The control parameter pairs will be helpful to maintain balance between exploration and exploitation of mutation strategies used in the strategy pool. The following combinations of control parameters are used in RCPDE algorithm.

1. F=1.0; CR=0.1 **[**120**]**
2. F=0.8; CR=0.2 **[**120**]**
3. F=0.7; CR=0.5 [177,182]
4. F=0.5; CR=0.9 **[**183**]**

6.3 Pseudocode of RCPDE

Figure 6.1 shows the pseudocode of Random controlled pool base Differential Evolution algorithm. The pseudocode contains an implementation flow of RCPDE algorithm that starts with parameter pool and strategy candidate pool initialization. After a random initialization of DE population numbers; fitness value of each population member is calculated. Then population members are evolved after selecting mutation strategy and control parameter values DE algorithm operators mutation, crossover and selection are applied. After evolutionary process the optimal solution is obtained. The proposed mutation variant RCPDE is also implemented through computer simulation.

The strategy candidate pool: “rand/1/bin”, “rand/2/bin”, “current-to-rand/1”, “rand to best /1/bin”, “TSDE/bin”
The parameter pool: [F=1.0, Cr=0.1], [F=0.8, Cr=0.2], [F=0.5, Cr=0.9], [F=0.7, Cr=0.5]

1. Generate the initial population $P_G = \{X_{1,G}, \ldots\ldots X_{NP,G}\}$ for generation G=0, randomly initialize each population member $X_{i,G} = \{x^1_{i,G} \ldots\ldots\ldots x^D_{i,G}\}$ where i = 1…………NP
2. Randomly initialize control parameter memory from control parameter pool and strategy memory from strategy pool for each population member $X_{i,G} = \{x^1_{i,G} \ldots\ldots\ldots x^D_{i,G}\}$ where i = 1…………NP.

3. FOR i = 1 to NP
    Calculate fitness $f(X_{i,G})$ for each population member $X_{i,G}$ using parameter pool value and mutation strategy assigned in step-2.
  END FOR

4. WHILE the stopping criterion is not true
  Step 3.1 Vectors selection
    Select vectors to be used in the equation of mutation strategy ***S*** (given in equaions1-4) from current Population
  Step 3.2 Mutation Step
    FOR i = 1 to NP
      For the ith target vector $X_{i,G}$ generate a donor vector $V_{i,G} = \{v^1_{i,G} \ldots\ldots v^D_{i,G}\}$ with ith strategy $\boldsymbol{S_i}$ from memory strategy and ith control parameter $\boldsymbol{F_i}$ from control parameter memory.
    END FOR
  Step 3.3 Crossover Step
    FOR i = 1 to NP
      For the ith target vector $X_{i,G}$ generate a trial vector $U_{i,G} = \{u^1_{i,G} \ldots\ldots u^D_{i,G}\}$ with the specified crossover scheme using control parameter $\boldsymbol{CR_i}$ from control parameter memory.
    END FOR
  Step 3.4 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 $f(U_{i,G} \le f\left(X_{i,G}\right)$, THEN $X_{i,G+1} = U_{i,G}$, $f(X_{i,G}) = f\left(U_{i,G}\right)$
        IF $f(U_{i,G} \le f\left(X_{best,G}\right)$, THEN $X_{best,G+1} = U_{i,G}$, $f(X_{best,G}) = f\left(U_{i,G}\right)$
        END IF
      ELSE
        Strategy memory Updating
          Randomly select a *mutation strategy* ***S*** from *strategy pool* and update control parameter memory for ith population member

        Control Parameter memory Updating
          Randomly select a pair of *control parameters* (***F***, ***CR***) from *parameter pool* and update control parameter memory for ith population member

      END IF
    END FOR

  Step # 3.7 *increment generation number G=G+1*
Step 4. END WHILE

Figure 6.1 *Pseudocode of Random Controlled Pool base Differential Evolution algorithm*

6.4 RCPDE Flowchart

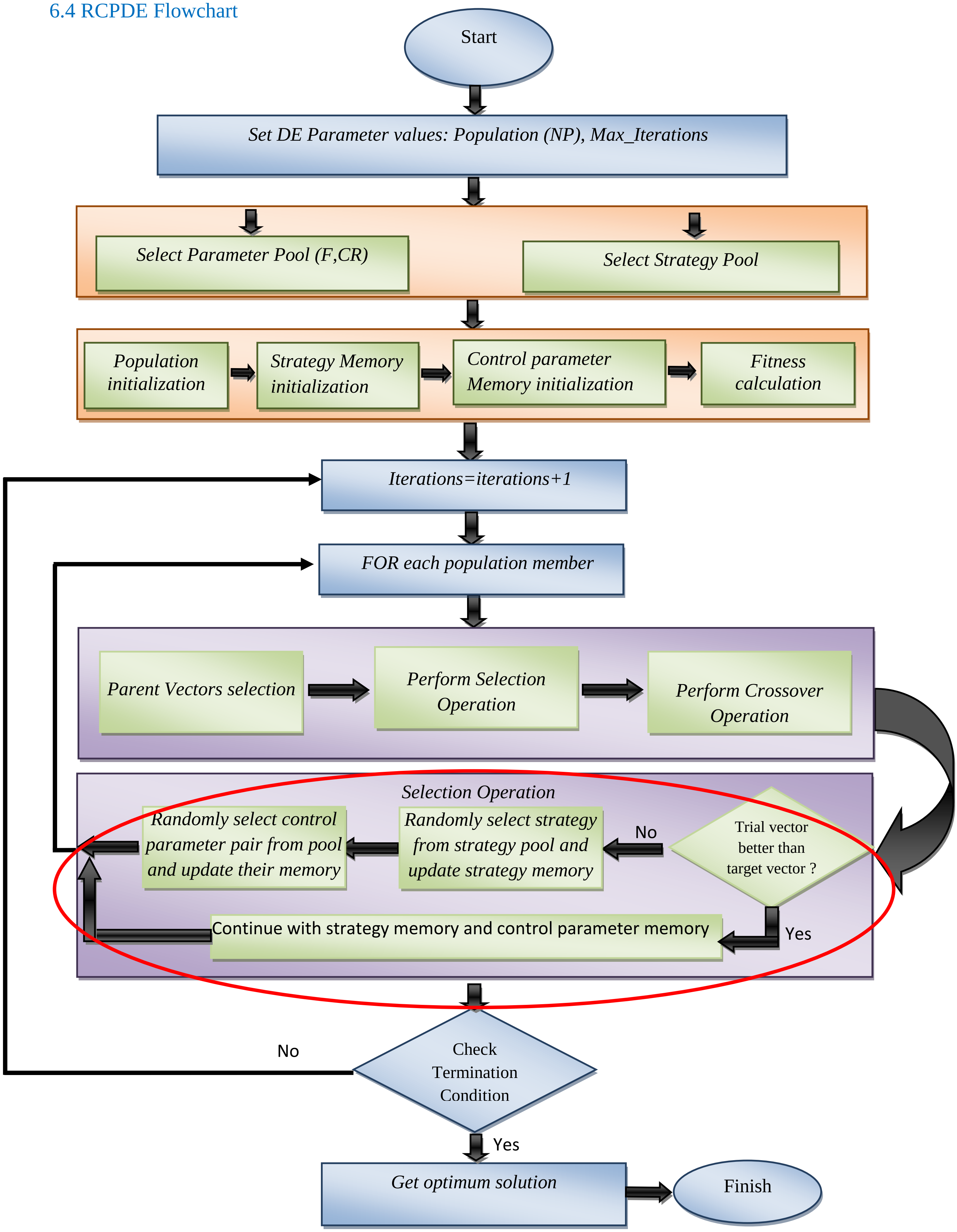


Figure 6.2: Flowchart of *Random controlled Pool base Differential Evolution algorithm*

6.5 RCPDE Parameter Setting

A set of N-dimensional test functions taken from **[**177**]** having varying characterizes like separable/non-separable, unimodal/multimodal are used to evaluate the performance of RCPDE and other DE variants. Experimental results are generated using 10D, 20D and 30D for the benchmark functions given in table-1 and table-2. Experimental results are generated by using control parameter population Size $N_p = 3D$ where D is dimension and dimensions are used as 10D, 20D and 30D with iterations 5000, 10000 and 15000 respectively. The average fitness value is calculated against average of 30 trials. Number of Function calls (NFC) are generated for maximum NFC 10^4 *DIM [178]. 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 all mutation strategies and all functions.

6.6 RCPDE Experimental Verification

Table.6.1– Number of function call results of functions ($f_1 - f_{15}$)

Function	DIM	DE	CoDE	EPSDE	jDE	RCPDE
f_1	10D	1.38E+02±8.43E+00	1.39E+02±5.97E+00	1.42E+02±5.84E+00	1.46E+02±7.98E+00	**1.21E+02±5.18E+00**
	20D	3.19E+02±9.04E+00	3.53E+02±5.64E+00	2.68E+02±9.30E+00	2.84E+02±1.25E+01	**2.07E+02±5.03E+00**
	30D	5.59E+02±1.58E+01	6.37E+02±8.17E+00	3.74E+02±8.28E+00	4.27E+02±1.89E+01	**2.74E+02±6.95E+00**
f_2	10D	1.39E+02±8.17E+00	1.34E+02±4.73E+00	1.42E+02±6.20E+00	1.40E+02±7.92E+00	**1.23E+02±5.70E+00**
	20D	3.44E+02±1.03E+01	3.70E+02±8.55E+00	2.92E+02±8.30E+00	3.08E+02±1.16E+01	**2.29E+02±6.60E+00**
	30D	6.29E+02±1.45E+01	6.94E+02±8.33E+00	4.24E+02±7.40E+00	4.79E+02±1.87E+01	**3.09E+02±7.36E+00**
f_3	10D	3.55E+02±1.24E+02	8.20E+02±3.63E+01	4.89E+02±2.73E+01	4.64E+02±3.23E+01	**2.92E+02±1.65E+01**
	20D	1.03E+03±6.89E+01	1.02E+04±2.94E+02	2.25E+03±8.78E+01	2.00E+03±1.10E+02	**8.71E+02±3.45E+01**
	30D	2.70E+03±9.73E+01	6.79E+04±1.34E+03	5.89E+03±1.81E+02	4.80E+03±2.89E+02	**1.79E+03±6.29E+01**
f_4	10D	-±-	9.85E+02±4.77E+01	9.96E+02±4.28E+01	8.43E+04±1.83E+04	**8.78E+02±1.12E+02**
	20D	9.94E+03±2.04E+03	3.32E+03±1.09E+02	2.31E+03±6.79E+01	1.22E+04±3.11E+03	**1.61E+03±4.84E+01**
	30D	4.26E+03±2.12E+02	7.47E+03±1.68E+02	3.82E+03±6.66E+01	1.36E+04±2.54E+03	**2.33E+03±4.89E+01**
f_5	10D	1.15E+03±1.56E+02	**2.54E+02±8.07E+00**	7.60E+02±5.60E+01	3.41E+02±3.07E+01	4.30E+02±2.54E+01
	20D	-±-	**7.36E+02±1.29E+01**	3.44E+03±1.48E+02	7.93E+02±6.90E+01	1.30E+03±4.81E+01
	30D	-±-	1.49E+03±1.90E+01	9.79E+03±3.55E+02	**1.27E+03±9.41E+01**	2.72E+03±7.22E+01
f_6	10D	4.89E+02±1.66E+02	4.58E+02±2.16E+01	1.23E+03±2.17E+02	**4.31E+02±6.07E+01**	7.90E+02±1.13E+02
	20D	5.19E+02±6.55E+01	6.50E+02±2.99E+01	4.98E+02±1.10E+02	4.65E+02±5.16E+01	**3.54E+02±5.82E+01**
	30D	8.11E+02±2.37E+01	1.03E+03±2.02E+01	5.75E+02±5.27E+01	6.07E+02±3.63E+01	**4.05E+02±3.18E+01**
f_7	10D	7.11E+01±6.62E+00	7.12E+01±4.56E+00	7.74E+01±5.92E+00	7.59E+01±6.99E+00	**6.56E+01±4.16E+00**
	20D	1.20E+02±8.84E+00	1.21E+02±9.13E+00	1.08E+02±6.87E+00	1.06E+02±9.29E+00	**8.61E+01±7.27E+00**
	30D	1.87E+02±1.26E+01	1.72E+02±8.85E+00	1.38E+02±8.97E+00	1.37E+02±1.39E+01	**1.01E+02±5.64E+00**
f_8	10D	8.47E+02±2.17E+02	**3.20E+02±6.91E+00**	9.11E+02±5.49E+01	4.26E+02±2.64E+01	5.05E+02±3.13E+01
	20D	-	**9.03E+02±1.54E+01**	3.95E+03±2.06E+02	9.49E+02±6.38E+01	1.43E+03±4.00E+01
	30D	-	1.78E+03±2.59E+01	1.12E+04±3.92E+02	**1.46E+03±8.01E+01**	2.87E+03±5.89E+01
f_9	10D	2.20E+02±1.11E+02	**1.04E+02±2.42E+01**	2.33E+02±1.63E+02	1.71E+02±7.01E+01	1.76E+02±6.60E+01
	20D	2.91E+02±1.44E+02	1.96E+02±1.99E+01	2.04E+02±5.47E+01	**1.83E+02±3.36E+01**	1.84E+02±6.93E+01
	30D	3.74E+02±9.56E+01	3.22E+02±2.02E+01	2.43E+02±3.58E+01	**2.30E+02±2.84E+01**	1.81E+02±3.01E+01
f_{10}	10D	2.70E+02±7.22E+01	5.71E+02±2.87E+01	3.89E+02±1.78E+01	3.69E+02±2.77E+01	**2.44E+02±1.44E+01**
	20D	8.71E+02±6.32E+01	4.02E+03±1.03E+02	1.76E+03±5.34E+01	1.42E+03±8.74E+01	**7.41E+02±2.81E+01**
	30D	2.36E+03±8.65E+01	1.22E+04±1.86E+02	4.55E+03±1.04E+02	3.17E+03±1.50E+02	**1.69E+03±4.78E+01**
f_{11}	10D	2.75E+02±8.61E+00	2.48E+02±5.79E+00	2.80E+02±8.45E+00	2.53E+02±1.04E+01	**2.37E+02±7.07E+00**
	20D	6.67E+02±1.82E+01	6.07E+02±6.60E+00	5.53E+02±1.17E+01	4.99E+02±1.81E+01	**4.30E+02±9.98E+00**
	30D	1.18E+03±2.83E+01	1.08E+03±1.03E+01	8.11E+02±1.09E+01	7.27E+02±2.35E+01	**5.90E+02±1.15E+01**
f_{12}	10D	9.31E+01±4.99E+00	8.91E+01±3.93E+00	9.48E+01±6.08E+00	9.49E+01±6.22E+00	**8.03E+01±7.18E+00**
	20D	2.13E+02±8.89E+00	2.29E+02±7.37E+00	1.79E+02±6.46E+00	1.86E+02±1.30E+01	**1.42E+02±6.45E+00**
	30D	3.77E+02±1.30E+01	4.13E+02±8.05E+00	2.53E+02±1.11E+01	2.75E+02±1.66E+01	**1.89E+02±1.01E+01**
f_{13}	10D	5.99E+01±6.15E+00	5.82E+01±3.60E+00	5.95E+01±5.03E+00	6.46E+01±7.65E+00	**5.24E+01±4.00E+00**
	20D	1.61E+02±1.04E+01	1.80E+02±5.42E+00	1.37E+02±5.99E+00	1.58E+02±1.26E+01	**1.04E+02±4.93E+00**
	30D	3.02E+02±1.36E+01	3.54E+02±9.58E+00	2.01E+02±8.05E+00	2.47E+02±1.46E+01	**1.47E+02±6.71E+00**
f_{14}	10D	**1.96E+03±2.03E+03**	4.19E+03±3.72E+03	1.92E+04±1.74E+04	8.18E+03±5.11E+03	2.28E+03±1.01E+03
	20D	1.88E+04±1.87E+04	**2.39E+03±2.48E+03**	1.43E+04±1.44E+04	8.31E+03±7.63E+03	8.61E+03±6.89E+03
	30D	4.18E+04±4.58E+04	**1.63E+03±1.76E+03**	7.91E+03±7.14E+03	6.99E+03±5.13E+03	5.66E+03±6.10E+03
f_{15}	10D	1.34E+02±8.21E+00	1.29E+02±6.21E+00	1.44E+02±8.40E+00	1.33E+02±1.07E+01	**1.19E+02±7.33E+00**
	20D	3.00E+02±1.43E+01	3.25E+02±7.28E+00	2.73E+02±9.81E+00	2.52E+02±1.29E+01	**1.99E+02±8.35E+00**
	30D	5.10E+02±1.80E+01	5.95E+02±1.14E+01	3.86E+02±1.32E+01	3.73E+02±1.53E+01	**2.62E+02±9.79E+00**

Table.6.2 – Number of function call results of functions ($f_{16} - f_{30}$)

Function	DIM	DE	CoDE	EPSDE	jDE	RCPDE
f_{16}	10D	-	-	-	-	-
	20D	1.23E+04±1.03E+04	8.99E+03±8.13E+03	2.43E+04±3.80E+04	**7.94E+03±6.33E+03**	1.45E+04±1.17E+04
	30D	8.61E+04±6.50E+04	**1.45E+04±1.02E+04**	4.74E+04±3.68E+04	1.53E+04±1.17E+04	3.75E+04±3.35E+04
f_{17}	10D	1.45E+02±9.38E+00	1.32E+02±5.15E+00	1.50E+02±8.17E+00	1.37E+02±8.84E+00	**1.27E+02±6.51E+00**
	20D	3.44E+02±1.76E+01	3.43E+02±6.33E+00	2.96E+02±1.04E+01	2.72E+02±1.55E+01	**2.25E+02±8.06E+00**
	30D	6.07E+02±2.11E+01	6.33E+02±8.67E+00	4.37E+02±1.10E+01	4.13E+02±1.46E+01	**3.12E+02±9.07E+00**
f_{18}	10D	2.58E+02±1.14E+01	2.57E+02±7.17E+00	2.64E+02±6.86E+00	2.64E+02±9.47E+00	**2.25E+02±5.99E+00**
	20D	5.81E+02±1.40E+01	6.43E+02±7.23E+00	4.84E+02±1.12E+01	5.14E+02±1.43E+01	**3.74E+02±6.65E+00**
	30D	1.01E+03±2.25E+01	1.15E+03±1.14E+01	6.76E+02±1.15E+01	7.47E+02±2.28E+01	**4.90E+02±9.47E+00**
f_{19}	10D	1.34E+02±8.63E+00	1.36E+02±4.71E+00	1.40E+02±5.29E+00	1.41E+02±7.82E+00	**1.19E+02±6.73E+00**
	20D	3.16E+02±1.00E+01	3.49E+02±6.91E+00	2.66E+02±6.90E+00	2.78E+02±1.19E+01	**2.03E+02±5.43E+00**
	30D	5.55E+02±1.88E+01	6.26E+02±9.98E+00	3.71E+02±7.61E+00	4.21E+02±1.60E+01	**2.72E+02±5.87E+00**
f_{20}	10D	1.96E+02±8.89E+00	1.92E+02±5.25E+00	1.99E+02±8.62E+00	2.00E+02±9.78E+00	**1.69E+02±7.48E+00**
	20D	4.41E+02±1.43E+01	4.87E+02±7.35E+00	3.68E+02±9.58E+00	3.93E+02±1.50E+01	**2.86E+02±8.23E+00**
	30D	7.67E+02±2.28E+01	8.75E+02±8.64E+00	5.15E+02±8.50E+00	5.78E+02±1.75E+01	**3.74E+02±7.75E+00**
f_{21}	10D	2.03E+02±8.05E+00	2.04E+02±5.82E+00	2.13E+02±8.29E+00	2.11E+02±6.95E+00	**1.81E+02±6.29E+00**
	20D	4.47E+02±1.12E+01	4.95E+02±8.36E+00	3.79E+02±6.76E+00	3.99E+02±1.57E+01	**2.97E+02±8.13E+00**
	30D	7.69E+02±2.28E+01	8.86E+02±9.99E+00	5.28E+02±8.90E+00	5.93E+02±2.54E+01	**3.87E+02±6.69E+00**
f_{22}	10D	2.75E+02±1.10E+02	2.51E+02±6.75E+00	2.33E+02±7.94E+00	2.73E+02±6.48E+01	**1.91E+02±8.19E+00**
	20D	5.29E+02±2.99E+01	6.63E+02±1.76E+01	4.68E+02±1.51E+01	5.38E+02±6.78E+01	**3.47E+02±1.57E+01**
	30D	9.08E+02±3.10E+01	1.21E+03±1.85E+01	7.13E+02±1.49E+01	8.40E+02±9.40E+01	**4.81E+02±1.93E+01**
f_{23}	10D	5.53E+02±5.72E+02	**2.33E+02±1.70E+02**	9.54E+02±8.86E+02	4.71E+02±3.40E+02	6.10E+02±6.54E+02
	20D	9.57E+02±9.39E+02	**2.49E+02±1.67E+02**	1.00E+03±9.27E+02	4.58E+02±3.66E+02	5.10E+02±3.74E+02
	30D	1.55E+03±1.04E+03	3.89E+02±2.72E+02	9.56E+02±7.02E+02	**3.73E+02±3.20E+02**	8.83E+02±5.84E+02
f_{24}	10D	2.09E+02±1.18E+01	**1.30E+02±2.76E+00**	1.70E+02±7.40E+00	1.77E+02±6.92E+00	1.65E+02±7.00E+00
	20D	4.55E+02±1.50E+01	**3.00E+02±5.07E+00**	3.21E+02±8.76E+00	3.31E+02±9.02E+00	3.02E+02±7.78E+00
	30D	7.70E+02±2.17E+01	4.84E+02±5.23E+00	4.56E+02±1.04E+01	4.78E+02±1.51E+01	**4.24E+02±8.23E+00**
f_{25}	10D	2.35E+02±1.40E+01	1.85E+02±3.63E+00	1.84E+02±6.49E+00	1.90E+02±7.14E+00	**1.78E+02±6.14E+00**
	20D	4.85E+02±1.95E+01	3.35E+02±4.05E+00	3.31E+02±8.27E+00	3.46E+02±8.83E+00	**3.12E+02±6.90E+00**
	30D	8.03E+02±2.23E+01	4.99E+02±4.41E+00	4.66E+02±1.12E+01	4.93E+02±1.34E+01	**4.37E+02±8.21E+00**
f_{26}	10D	4.19E+01±9.73E+00	**2.29E+01±3.57E+00**	3.77E+01±5.84E+00	2.76E+01±4.62E+00	3.35E+01±5.46E+00
	20D	6.78E+01±2.48E+01	**3.42E+01±3.94E+00**	5.30E+01±8.13E+00	3.92E+01±7.44E+00	4.44E+01±6.62E+00
	30D	1.12E+02±3.73E+01	**4.65E+01±5.19E+00**	6.75E+01±1.01E+01	4.87E+01±6.93E+00	6.21E+01±9.07E+00
f_{27}	10D	**4.17E+02±4.29E+01**	5.70E+02±5.85E+01	6.60E+02±1.49E+02	4.22E+02±5.33E+01	5.33E+02±1.25E+02
	20D	**1.08E+03±1.21E+02**	2.28E+03±1.08E+02	1.82E+03±3.40E+02	1.17E+03±1.68E+02	1.65E+03±6.46E+02
	30D	**1.83E+03±1.36E+02**	5.81E+03±1.01E+02	3.32E+03±6.63E+02	2.17E+03±3.97E+02	6.06E+03±3.35E+03
f_{28}	10D	**3.74E+02±2.36E+01**	-	6.92E+02±4.35E+01	4.47E+02±3.23E+01	5.35E+02±3.55E+01
	20D	**1.03E+03±4.33E+01**	-	1.99E+03±9.59E+01	1.09E+03±5.89E+01	1.26E+03±6.38E+01
	30D	1.98E+03±5.09E+01	-	3.73E+03±1.45E+02	**1.87E+03±1.05E+02**	2.13E+03±7.73E+01
f_{29}	10D	1.76E+01±1.47E+01	**1.49E+01±1.24E+01**	4.20E+01±4.46E+01	2.84E+01±1.99E+01	3.69E+01±2.68E+01
	20D	8.97E+00±8.00E+00	**7.40E+00±6.52E+00**	1.67E+01±1.12E+01	1.29E+01±1.49E+01	1.85E+01±2.35E+01
	30D	9.07E+00±7.85E+00	**5.20E+00±3.59E+00**	9.77E+00±8.95E+00	8.97E+00±1.02E+01	2.76E+01±2.01E+01
f_{30}	10D	-	-	-	-	-
	20D	5.69E+02±4.39E+02	**1.31E+01±3.92E+00**	4.46E+01±2.05E+01	3.85E+01±1.01E+01	3.33E+01±9.40E+00
	30D	1.43E+01±1.11E+01	**4.87E+00±1.43E+00**	7.67E+00±3.81E+00	8.47E+00±5.07E+00	7.67E+00±3.82E+00

Table.6.3 –Average Fitness results of functions (f_1 - f_{15})

Function	DIM	DE	CoDE	EPSDE	jDE	RCPDE
f_1	10D	4.96E-82±2.67E-81	5.82E-231±0.00E+00	8.42E-228±0.00E+00	9.78E-228±0.00E+00	**2.11E-270±0.00E+00**
	20D	9.38E-212±0.00E+00	5.34E-191±0.00E+00	3.76E-254±0.00E+00	1.76E-241±0.00E+00	**0.00E+00±0.00E+00**
	30D	3.92E-187±0.00E+00	9.66E-163±0.00E+00	5.14E-280±0.00E+00	3.33E-250±0.00E+00	**0.00E+00±0.00E+00**
f_2	10D	1.30E-113±7.02E-113	2.57E-257±0.00E+00	5.72E-245±0.00E+00	5.29E-244±0.00E+00	**5.97E-286±0.00E+00**
	20D	8.85E-222±0.00E+00	1.68E-203±0.00E+00	8.46E-264±0.00E+00	3.76E-250±0.00E+00	**0.00E+00±0.00E+00**
	30D	1.11E-191±0.00E+00	1.79E-170±0.00E+00	3.01E-285±0.00E+00	2.78E-255±0.00E+00	**0.00E+00±0.00E+00**
f_3	10D	7.91E-04±2.70E-03	1.50E-47±4.71E-47	1.73E-82±5.72E-82	3.79E-80±1.49E-79	**2.11E-140±4.35E-140**
	20D	1.62E-80±3.35E-80	1.77E-05±1.03E-05	2.77E-36±2.85E-36	6.68E-41±2.23E-40	**5.48E-96±1.71E-95**
	30D	1.72E-46±3.67E-46	7.99E+01±1.77E+01	1.44E-19±1.70E-19	2.84E-23±6.52E-23	**2.51E-72±7.25E-72**
f_4	10D	5.77E+00±1.48E+00	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	1.67E+00±1.35E+00	1.33E-01±7.16E-01
	20D	1.34E-02±2.41E-02	1.50E-29±4.10E-29	**0.00E+00±0.00E+00**	5.41E-01±8.49E-01	2.68E-30±1.18E-29
	30D	1.33E-01±7.16E-01	3.63E-19±4.58E-19	**0.00E+00±0.00E+00**	3.20E-01±9.93E-01	1.33E-01±7.16E-01
f_5	10D	2.40E+00±1.59E+00	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	1.99E-01±4.74E-01	**0.00E+00±0.00E+00**
	20D	5.89E+00±2.51E+00	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	6.63E-02±2.48E-01	**0.00E+00±0.00E+00**
	30D	8.91E+00±2.33E+00	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	9.95E-02±2.98E-01	**0.00E+00±0.00E+00**
f_6	10D	3.34E-02±3.10E-02	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	6.65E-03±9.66E-03	**0.00E+00±0.00E+00**
	20D	1.73E-03±3.55E-03	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**2.47E-04±1.33E-03**
	30D	1.15E-03±3.69E-03	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	2.47E-04±1.33E-03	**0.00E+00±0.00E+00**
f_7	10D	2.53E-11±7.80E-11	**0.00E+00±0.00E+00**	4.56E-294±0.00E+00	3.20E-60±1.72E-59	**0.00E+00±0.00E+00**
	20D	1.48E-28±7.61E-28	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
	30D	1.16E-109±6.22E-109	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
f_8	10D	1.59E-01±9.48E-02	**6.26E-13±6.20E-13**	2.31E-12±2.48E-12	3.52E-02±5.90E-02	3.08E-12±5.02E-12
	20D	1.57E-01±7.59E-02	**1.17E-13±1.19E-13**	1.31E-12±1.19E-12	2.07E-02±3.74E-02	5.40E-13±5.23E-13
	30D	1.73E-01±4.67E-02	**1.02E-13±7.57E-14**	1.01E-12±6.57E-13	9.62E-03±2.45E-02	2.43E-13±2.88E-13
f_9	10D	3.10E-06±7.03E-06	3.17E-08±2.76E-08	1.80E-07±1.21E-07	**1.10E-08±3.40E-08**	1.81E-07±2.14E-07
	20D	1.92E-07±2.88E-07	1.47E-08±1.58E-08	5.58E-08±4.15E-08	**1.15E-08±1.86E-08**	3.62E-08±2.54E-08
	30D	8.55E-08±8.50E-08	1.10E-08±1.41E-08	3.29E-08±3.46E-08	**1.03E-08±9.20E-09**	3.36E-08±3.00E-08
f_{10}	10D	9.81E-07±4.30E-06	4.04E-65±8.35E-65	5.49E-98±1.73E-97	5.91E-100±3.10E-99	**1.59E-156±3.80e-311**
	20D	1.11E-92±2.12E-92	7.84E-18±4.46E-18	1.88E-46±3.49E-46	6.67E-56±1.73E-55	**4.88E-111±1.22E-110**
	30D	1.09E-52±2.92E-52	1.25E-07±3.85E-08	3.64E-27±2.97E-27	1.67E-37±7.58E-37	**2.44E-77±5.99E-77**
f_{11}	10D	4.94E-119±1.11E-118	6.56E-129±1.86E-128	1.21E-115±1.72E-115	3.58E-127±8.54E-127	**5.73E-137±9.90E-137**
	20D	1.43E-104±2.22E-104	3.32E-110±3.72E-110	2.70E-125±2.37E-125	1.99E-137±7.76E-137	**1.49E-162±4.94e-324**
	30D	1.59E-90±3.04E-90	4.26E-96±2.35E-96	1.11E-133±7.66E-134	1.65E-143±2.59E-143	**2.15E-184±0.00E+00**
f_{12}	10D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
	20D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
	30D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
f_{13}	10D	4.21E-09±2.26E-08	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	2.63E-307±0.00E+00	**0.00E+00±0.00E+00**
	20D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
	30D	**0.00E+00±0.00E+00**	6.11E-271±0.00E+00	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
f_{14}	10D	2.66E-04±3.52E-04	**7.26E-06±6.97E-06**	4.46E-05±4.65E-05	2.25E-05±3.22E-05	4.60E-05±3.67E-05
	20D	3.55E-05±3.05E-05	**3.29E-06±3.70E-06**	1.43E-05±1.32E-05	7.56E-06±6.18E-06	9.07E-06±8.30E-06
	30D	2.33E-05±1.87E-05	**9.53E-07±7.90E-07**	9.18E-06±1.26E-05	4.16E-06±2.93E-06	4.29E-06±4.72E-06
f_{15}	10D	**3.27E-31±1.31E-46**	**3.27E-31±1.31E-46**	**3.27E-31±1.31E-46**	**3.27E-31±1.31E-46**	**3.27E-31±1.31E-46**
	20D	**1.63E-31±6.57E-47**	**1.63E-31±6.57E-47**	**1.63E-31±6.57E-47**	**1.63E-31±6.57E-47**	**1.63E-31±6.57E-47**
	30D	**1.09E-31±8.76E-47**	**1.09E-31±8.76E-47**	**1.09E-31±8.76E-47**	**1.09E-31±8.76E-47**	**1.09E-31±8.76E-47**

Table.6.4 –Average Fitness results of functions (f_{16} - f_{30})

Function	DIM	DE	CoDE	EPSDE	jDE	RCPDE
f_{16}	10D	**8.53E+01±1.85E-02**	**8.53E+01±7.11E-14**	**8.53E+01±7.11E-14**	**8.53E+01±7.11E-14**	**8.53E+01±7.11E-14**
	20D	6.03E-05±4.67E-05	**8.32E-06±6.50E-06**	3.05E-05±3.52E-05	8.58E-06±1.19E-05	1.75E-05±1.88E-05
	30D	6.08E-05±7.61E-05	**1.28E-05±1.21E-05**	3.72E-05±4.52E-05	1.64E-05±1.60E-05	2.85E-05±2.78E-05
f_{17}	10D	3.28E-08±1.76E-07	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
	20D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
	30D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
f_{18}	10D	1.16E-48±6.27E-48	3.22E-225±0.00E+00	2.30E-221±0.00E+00	1.62E-220±0.00E+00	**5.51E-265±0.00E+00**
	20D	1.54E-206±0.00E+00	1.92E-185±0.00E+00	3.68E-249±0.00E+00	4.77E-231±0.00E+00	**0.00E+00±0.00E+00**
	30D	2.23E-181±0.00E+00	4.24E-157±1.67e-313	3.75E-274±0.00E+00	3.14E-244±0.00E+00	**0.00E+00±0.00E+00**
f_{19}	10D	8.98E-21±4.83E-20	3.85E-230±0.00E+00	4.75E-227±0.00E+00	6.55E-225±0.00E+00	**2.13E-268±0.00E+00**
	20D	6.58E-212±0.00E+00	1.36E-190±0.00E+00	1.74E-254±0.00E+00	1.17E-240±0.00E+00	**0.00E+00±0.00E+00**
	30D	4.42E-187±0.00E+00	7.18E-163±0.00E+00	6.03E-280±0.00E+00	9.89E-250±0.00E+00	**0.00E+00±0.00E+00**
f_{20}	10D	2.86E-06±1.54E-05	7.82E-228±0.00E+00	1.48E-224±0.00E+00	1.09E-223±0.00E+00	**1.56E-267±0.00E+00**
	20D	1.71E-209±0.00E+00	3.85E-188±0.00E+00	8.52E-252±0.00E+00	3.06E-237±0.00E+00	**0.00E+00±0.00E+00**
	30D	2.62E-184±0.00E+00	3.97E-160±2.71e-319	4.95E-277±0.00E+00	2.46E-247±0.00E+00	**0.00E+00±0.00E+00**
f_{21}	10D	5.08E-16±2.72E-15	9.60E-228±0.00E+00	6.45E-224±0.00E+00	2.91E-224±0.00E+00	**1.57E-265±0.00E+00**
	20D	6.27E-210±0.00E+00	2.51E-188±0.00E+00	6.91E-251±0.00E+00	7.79E-236±0.00E+00	**0.00E+00±0.00E+00**
	30D	2.08E-183±0.00E+00	6.70E-160±8.42E+00	6.84E-277±0.00E+00	3.67E-246±0.00E+00	**0.00E+00±0.00E+00**
f_{22}	10D	2.26E-03±1.05E-02	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	3.85E-31±6.23E-31	**0.00E+00±0.00E+00**
	20D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	4.29E-31±3.97E-31	**0.00E+00±0.00E+00**
	30D	4.81E-32±1.44E-31	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	7.87E-31±5.54E-31	**0.00E+00±0.00E+00**
f_{23}	10D	1.11E-05±1.39E-05	2.46E-07±2.44E-07	1.40E-06±1.22E-06	**3.79E-08±1.60E-07**	1.29E-06±1.75E-06
	20D	9.42E-07±8.89E-07	2.11E-07±2.22E-07	6.13E-07±6.51E-07	**1.46E-07±1.81E-07**	4.26E-07±4.40E-07
	30D	9.18E-07±1.02E-06	**1.49E-07±1.18E-07**	5.33E-07±4.78E-07	1.58E-07±1.52E-07	3.73E-07±4.44E-07
f_{24}	10D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
	20D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
	30D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
f_{25}	10D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
	20D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
	30D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
f_{26}	10D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
	20D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
	30D	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
f_{27}	10D	**1.69E-15±2.92E-16**	1.80E-15±2.47E-16	1.97E-15±1.39E-16	1.74E-15±2.63E-16	1.94E-15±1.64E-16
	20D	3.97E-15±2.13E-16	**1.26E-15±5.74E-16**	4.11E-15±9.88E-17	3.78E-15±2.49E-16	4.14E-15±5.18E-17
	30D	6.11E-15±1.39E-16	**2.21E-16±2.20E-16**	6.22E-15±3.73E-17	6.00E-15±2.50E-16	4.38E-09±1.94E-08
f_{28}	10D	**6.16E-80±1.88E-79**	5.13E-01±9.66E-02	1.26E-41±4.03E-41	3.63E-66±8.41E-66	1.87E-53±7.25E-53
	20D	**5.62E-59±9.63E-59**	3.93E+00±4.20E-01	6.77E-30±9.78E-30	5.44E-56±1.18E-55	1.50E-46±3.23E-46
	30D	2.41E-46±3.59E-46	9.12E+00±7.90E-01	2.30E-24±3.66E-24	**1.02E-50±2.76E-50**	2.88E-43±6.30E-43
f_{29}	10D	1.19E-07±5.27E-07	7.41E-08±1.58E-07	3.73E-10±5.25E-10	**2.89E-29±3.00E-29**	2.22E-08±4.18E-08
	20D	2.46E-09±1.15E-08	1.32E-09±4.94E-09	7.96E-11±1.71E-10	**2.70E-29±4.24E-29**	8.50E-10±1.53E-09
	30D	8.18E-10±2.30E-09	1.54E-10±4.15E-10	1.19E-11±1.95E-11	**3.75E-29±4.91E-29**	4.04E-10±8.71E-10
f_{30}	10D	2.31E-04±2.89E-05	**2.08E-04±2.62E-20**	**2.08E-04±2.27E-20**	2.09E-04±4.68E-06	**2.08E-04±1.40E-20**
	20D	2.58E-08±5.01E-09	**1.90E-08±3.20E-24**	**1.90E-08±6.28E-11**	**1.90E-08±5.30E-24**	**1.90E-08±5.23E-23**
	30D	2.05E-12±4.54E-13	1.29E-12±3.57E-28	3.80E-12±2.99E-13	**1.29E-12±1.32E-27**	**1.29E-12±2.88E-20**

6.7 Discussion on Experimental Results

The experimental results of number of function call performance parameter are generated using the setting discussed in section 6.6 of this chapter. The result of performance parameters are large enough to be reported in a single table and so are divided into multiple tables. Tables 6.1-2 contains NFC values and their corresponding standard deviation where the best values are reported as boldfaces. Experimental results of DE, EPSDE, CoDE, jDE and RCPDE are obtained over varying nature multidimensional functions. The proposed RCPDE has dominating NFC performance for separable functions *Sphere model,Axis parallel hyperellipsoid*, *Step function*, *De Jong's function 4 (no noise)*, *Levy and Montalvo Problem*, *Cosine Mixture*, *Cigar, Function '15'*, *Tablet Function*, *Ellipse Function*, *Schewel;* for non separable*: Schwefel's problem*-*1.2*, *Rosenbrock's valley*, *Griewank's function*, *Sum of different power*, *Zakharov function*, *Schwefel's problem 2.22*, *Mishra-1 global optimization*, *Mishra-2 global optimization;* for unimodal functions *Axis parallel hyperellipsoid*, *Schwefel's problem 1.2*, *Rosenbrock's valley*, *Schwefel's problem 2.22*, *Step function*, *De Jong's function 4 (no noise)*, *Ellipse Function*, *Tablet Function;* for multimodal functions *Sphere model, Griewank's function*, *Sum of different power*, *Zakharov function*, *Levy and Montalvo Problem*, *Cosine Mixture*, *Cigar*, *Function '15'*, *Schewel,Mishra-1 global optimization*, *Mishra-2 global optimization.* DE Algorithm has better performance for separable functions *Alpine function(10D)*, *Quintic global optimization problem*, *Stochastic global optimization problem(10D, 20D)*; and for multimodal functions *Alpine function(10D)*, *Quintic global optimization problem*, *Stochastic global optimization problem(10D, 20D)*. The CoDE algorithm has better performance for separable functions *Rastrigin's function(10D,20D)*, *Levy function(10D), Alpine function(20D,30D), Neumaier-2 Problem(30D), Deflected Corrugated Spring(10D,20D)*, *MultiModal global optimization problem;* for non-separable function *Ackley's path function(10D,20D)*, *Stretched-V global optimization problem*, *XinSheYang(20D,30D);* for unimodal functions *Neumaier-2 Problem(30D)* and for multimodal functions *Rastrigin's function(10D,20D)*, *Ackley's path function(10D,20D)*, *Levy function(10D)*, *Alpine function(20D,30D)*, *Deflected Corrugated Spring(10D,20D)*, *MultiModal global optimization problem*, *Stretched-V global optimization problem*, *XinSheYang(20D,30D).* The jDE algorithm has better NFC performance for Separable functions *Rastrigin's function(30D)*, *Levy function(20D, 30D)*, *Neumaier-2 Problem*, *Neumaier-2 Problem(20D)*, *Deflected Corrugated Spring(30D)*, *Stochastic global optimization problem(30D); for* Non-Separable functions *Ackley's path function(30D)*; for Unimodal functions *Neumaier-2 Problem(20D)* and for Multimodal functions *Rastrigin's function(30D)*, *Ackley's path function(30D), Levy function(20D, 30D), Deflected Corrugated Spring(30D), Stochastic global optimization problem(30D).*

Average fitness results of DE, EPSDE, CoDE, jDE and RCPDE are reported in table-6.3 and table-6.4. From average fitness results it can be observed that the fitness performance of proposed RCPDE has better fitness performance in most of the cases. RCPDE has better average fitness performance for separable functions f_1, f_2, f_3, f_5, f_{12}, f_{13}, f_{15}, $f_{16}(10D)$, f_{17}, f_{18}, f_{19}, f_{20}, f_{21}, f_{22}, f_{26}; for non-separable functions f_6, f_7, f_{10}, f_{11}, f_{24}, f_{25}, f_{30}; for unimodal functions f_2, f_3, f_{11}, f_{12}, f_{13}, $f_{16}(10D)$, f_{20}, f_{21}, f_{22} and multimodal functions f_1, f_5, f_6, f_7, f_{10}, f_{15}, f_{17}, f_{18}, f_{19}, f_{24}, f_{25}, f_{26}, f_{30}. The DE algorithm average fitness results are better for separable functions f_{12}, $f_{13}(20D,30D)$, f_{15}, $f_{16}(10D)$, $f_{17}(20D,30D)$, $f_{22}(20D)$, f_{26}, $f_{27}(10D)$, $f_{28}(10D,20D)$; for non separable function f_{24}, f_{25}; for unimodal functions f_{12}, $f_{13}(20D,30D)$, $f_{16}(10D)$ and multimodal functions f_{15}, $f_{17}(20D,30D)$, $f_{22}(20D)$, f_{24}, f_{25}, f_{26}, $f_{27}(10D)$, $f_{28}(10D,20D)$. The jDE algorithm has better average fitness performance for separable functions f_9, f_{12}, $f_{13}(20D,30D)$, f_{15}, $f_{16}(10D)$, f_{17}, $f_{23}(10D,20D)$, f_{26}, $f_{28}(30D)$; for non separable functions $f_6(20D)$, $f_7(20D,30D)$, f_{24}, f_{25}, f_{29}, $f_{30}(20D,30D)$; for unimodal functions f_{12}, $f_{13}(20D,30D)$, $f_{16}(10D)$ and for multimodal functions $f_6(20D)$, $f_7(20D,30D)$, f_9, f_{15}, f_{17}, $f_{23}(10D,20D)$, f_{24}, f_{25}, f_{26}, $f_{28}(30D)$, f_{29}, $f_{30}(20D,30D)$. The CoDE algorithm has better average fitness performance for separable functions f_5, f_{12}, $f_{13}(10D,20D)$, f_{14}, f_{15}, f_{16}, f_{17}, f_{22}, $f_{23}(30D)$, f_{26}, $f_{27}(20D,30D)$; for non separable functions $f_4(10D)$, f_6, f_7, f_8, f_{24}, f_{25}, $f_{30}(10D,20D)$; for unimodal functions $f_4(10D)$, f_{12}, $f_{13}(10D,20D)$, f_{16} and for multimodal functions f_5, f_6, f_7, f_8, f_{14}, f_{15}, f_{17}, f_{22}, $f_{23}(30D)$, f_{24}, f_{25}, f_{26}, $f_{27}(20D,30D)$, $f_{30}(10D,20D)$. The EPSDE algorithm has better average fitness performance for separable functions f_5, f_{12}, f_{13}, f_{15}, $f_{16}(10D)$, f_{17}, f_{22}, f_{26}; for non-separable functions f_4, f_6, $f_7(20D,30D)$, f_{24}, f_{25}, $f_{30}(10D,20D)$; for unimodal functions f_4, f_{12}, f_{13}, $f_{16}(10D)$ and for multimodal functions f_5 ,f_6, $f_7(20D,30D)$, f_{15}, f_{17}, f_{22}, f_{24}, f_{25}, f_{26}, $f_{30}(10D,20D)$. The overall results of NFC and average fitness can be summarized that the performance of proposed RCPDE is dominating for unimodal, multimodal separable and non-separable functions.

6.8 Convergence Graphs of RCPDE Algorithm

This section contains logarithmic convergence graphs of selected functions showing iterations horizontally and performance vertically. Convergence graphs of DE, EPSDE, CoDE, jDE, RCPDE generated for 10D average fitness values. It can be observed from convergence graphs that the performance of proposed RCPDE is better than DE, EPSDE, CoDE and jDE. The proposed RCPDE will prove to be a significant addition to DE literature.

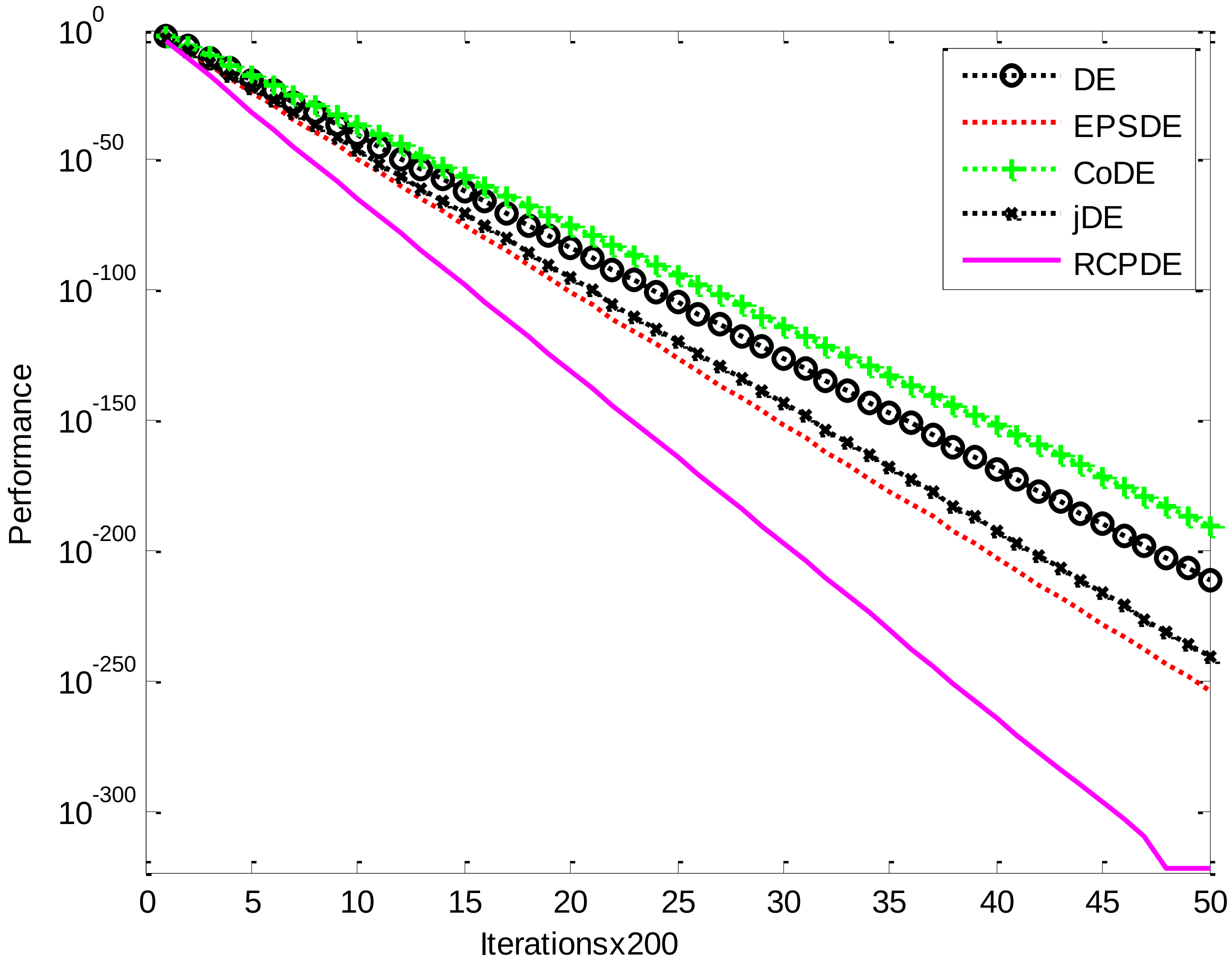


Figure 6.3- 20D average fitness logarithmic convergence graphs for f_1 showing number of iterations horizontally and average fitness vertically

Figure 6.3 contains an average fitness convergence graph of proposed RCPDE and other state of the art DE algorithms for sphere model function (f_1). This figure depicts that RCPDE has quick convergence from starting iteration till final iteration among all other algorithms. Only RCPDE reaches at optimal value 0 within the given iterations. The performance of CoDE algorithm is similar to other algorithms in the starting iterations, however it slows down in the subsequent iterations. It is clear from this figure that only RCPDE reaches at optimal value 0 The performance of EPSDE and jDE is similar in early iterations; however, EPSDE has fast convergence in the succeeding iterations.

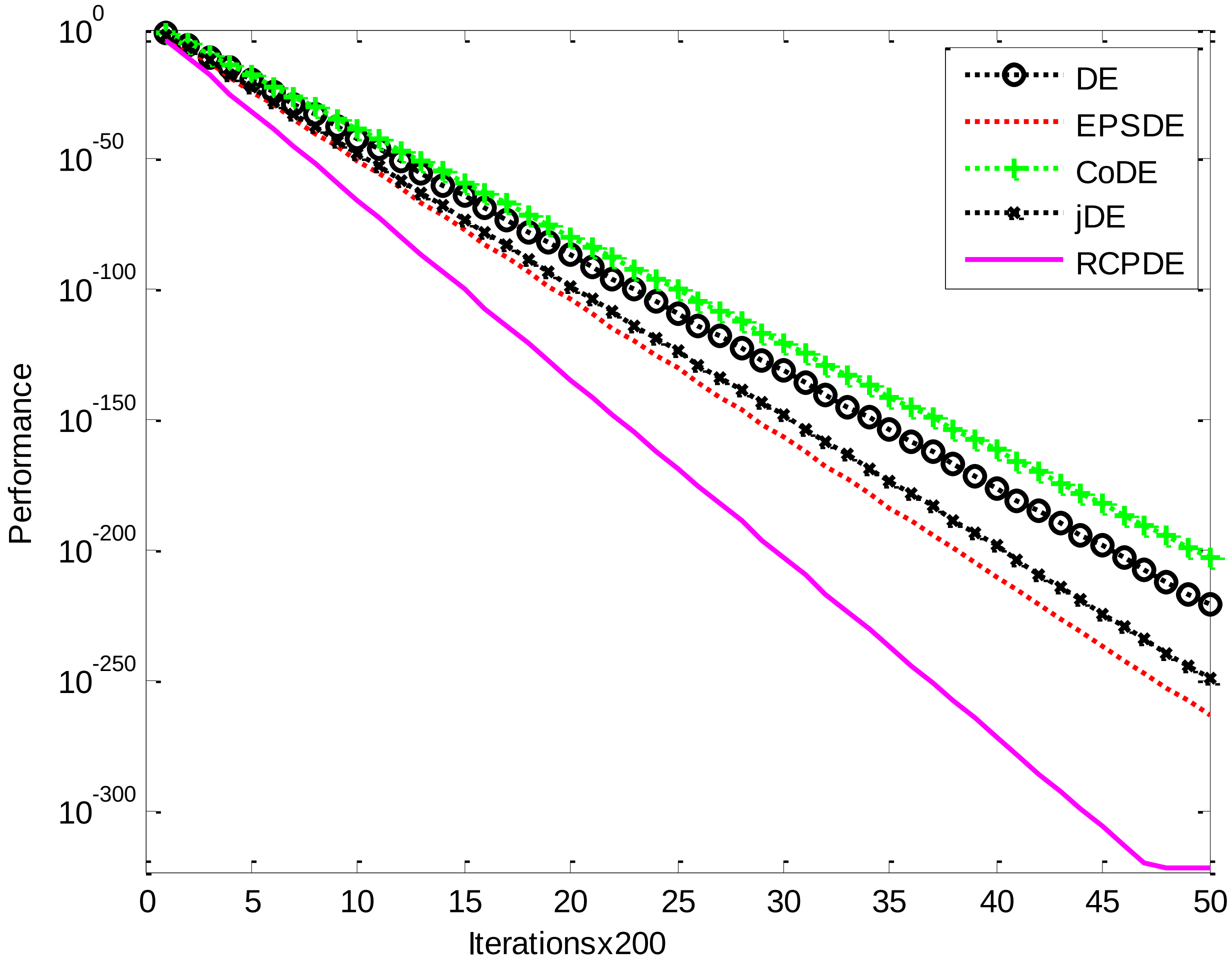


Figure 6.4- 20D average fitness logarithmic convergence graphs for f_2 showing number of iterations horizontally and average fitness vertically

The average fitness value of RCPDE and other DE algorithm for Axis Parallel Hyperellipsoid (f_2) optimization problem are graphically presented in Figure 6.4. This figure clearly indicates that CoDE has the worst performance among all algorithms. RCPDE maintains quick convergence from starting iterations till final iteration and reaches at optimal value within given iterations. The performance of EPSDE is similar to jDE in early iterations, however, EPSDE convergence improves in subsequent iterations and secure second best algorithm among all algorithms.

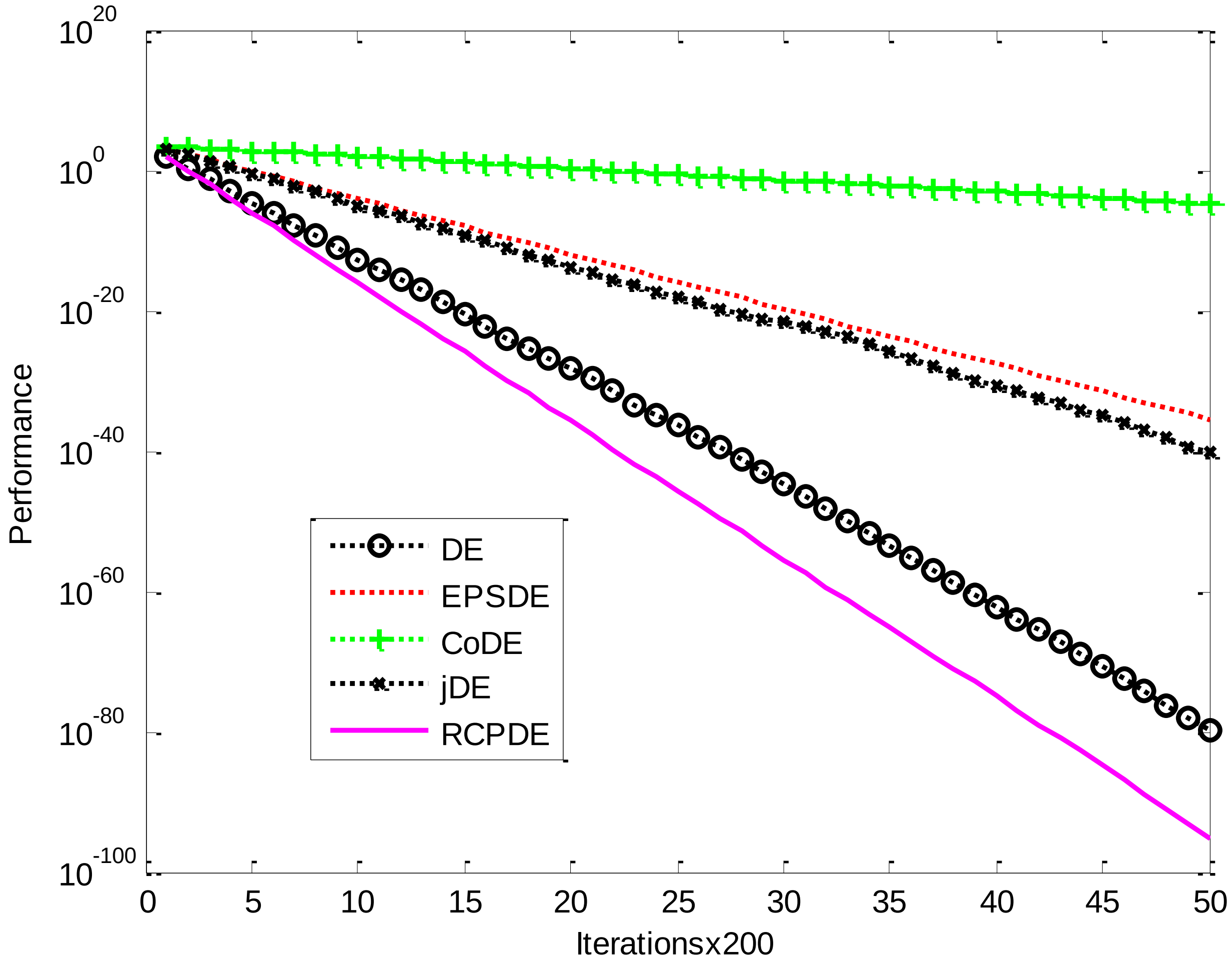


Figure 6.5- 20D average fitness logarithmic convergence graphs for f_3 showing number of iterations horizontally and average fitness vertically

The average fitness performance of RCPDE and other state of the art algorithms for Schwefel's problem-1.2 (f_3) are contained in Figure 6.5. This figure illustrates that CoDE has the worst performance among all other DE algorithms till final iteration. RCPDE has quick convergence speed than all other algorithms in the specified iterations. The standard DE algorithm has the second best convergence speed for this problem. jDE and EPSDE have similar performance in the starting iterations, however, jDE convergence improves in the succeeding iterations.

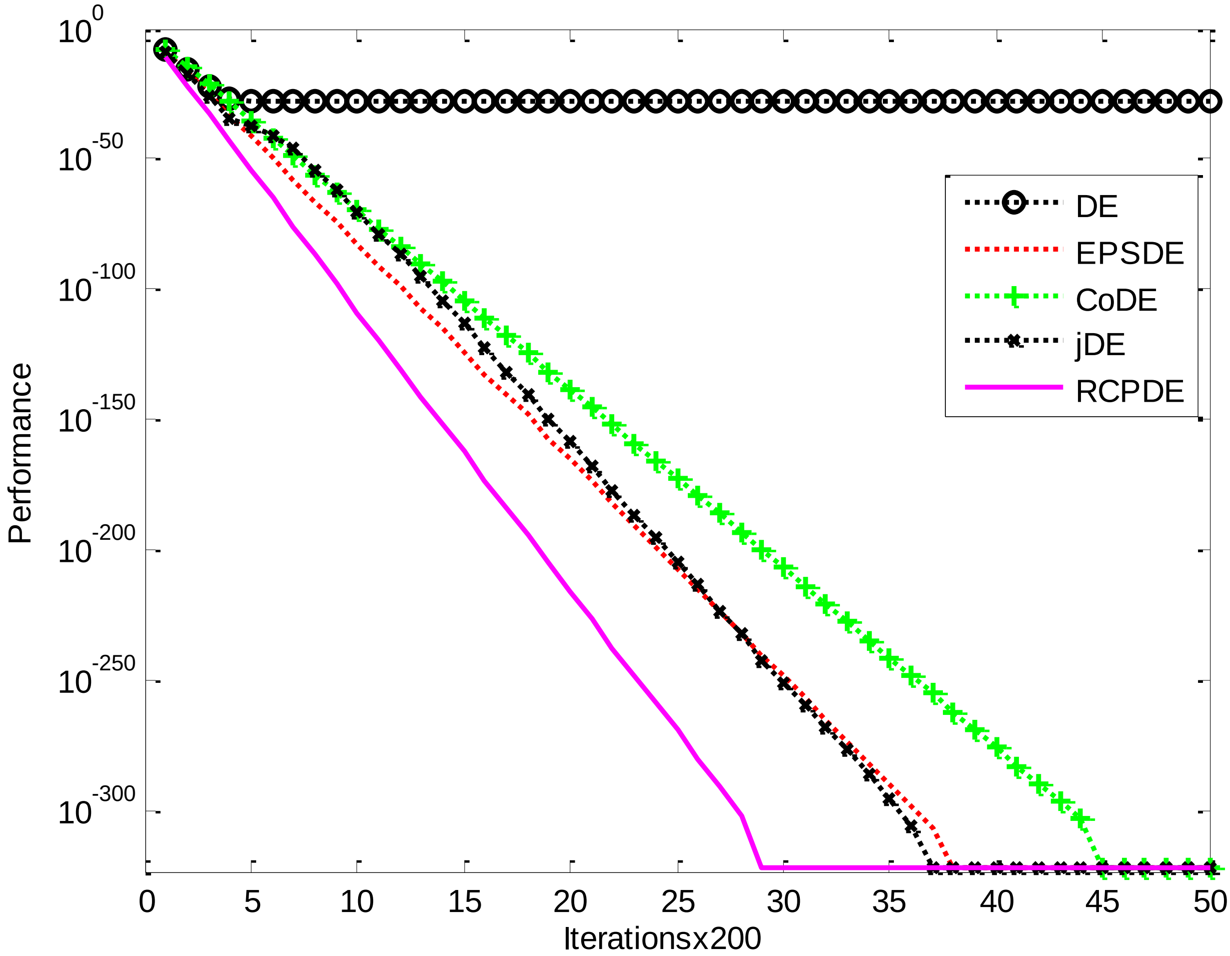


Figure 6.6- 20D average fitness logarithmic convergence graphs for f_7 showing number of iterations horizontally and average fitness vertically

Average fitness value convergence performance of RCPDE and other state of the art algorithms for Sum of different power function (f_7) is shown in Figure 6.6. Convergence speed of DE algorithm is worst among all algorithms. RCPDE, jDE, EPSDE and CoDE achieves value 0 within the specified iterations, however, among all these algorithms RCPDE has quick convergence since it reaches at optimal value in less iterations while jDE, EPSDE and CoDE reaches at 0 in more iterations.

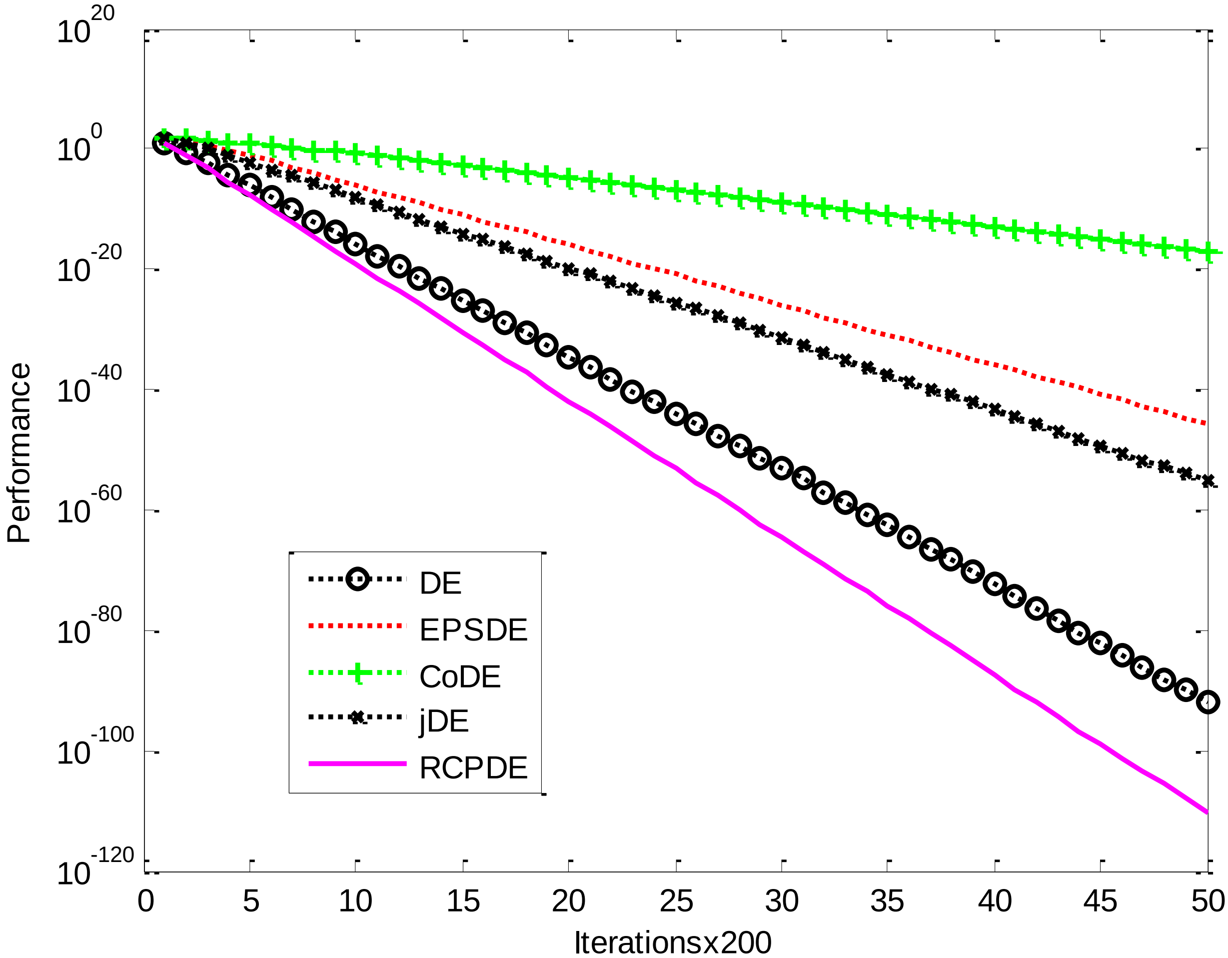


Figure 6.7- 20D average fitness logarithmic convergence graphs for f_{10} showing number of iterations horizontally and average fitness vertically

Average fitness convergence Performance of RCPDE and other algorithms is presented graphically in Figure 6.7 for Zakharov function (f_{10}). This figure depicts that CoDE has slowest performance among all other algorithms that achieves value 7.84E-18 in the specified iterations, however, the convergence speed of RCPDE is fastest among all other algorithms. It is important to note that the performance standard DE algorithm is better than jDE, CoDE and EPSDE for this problem.

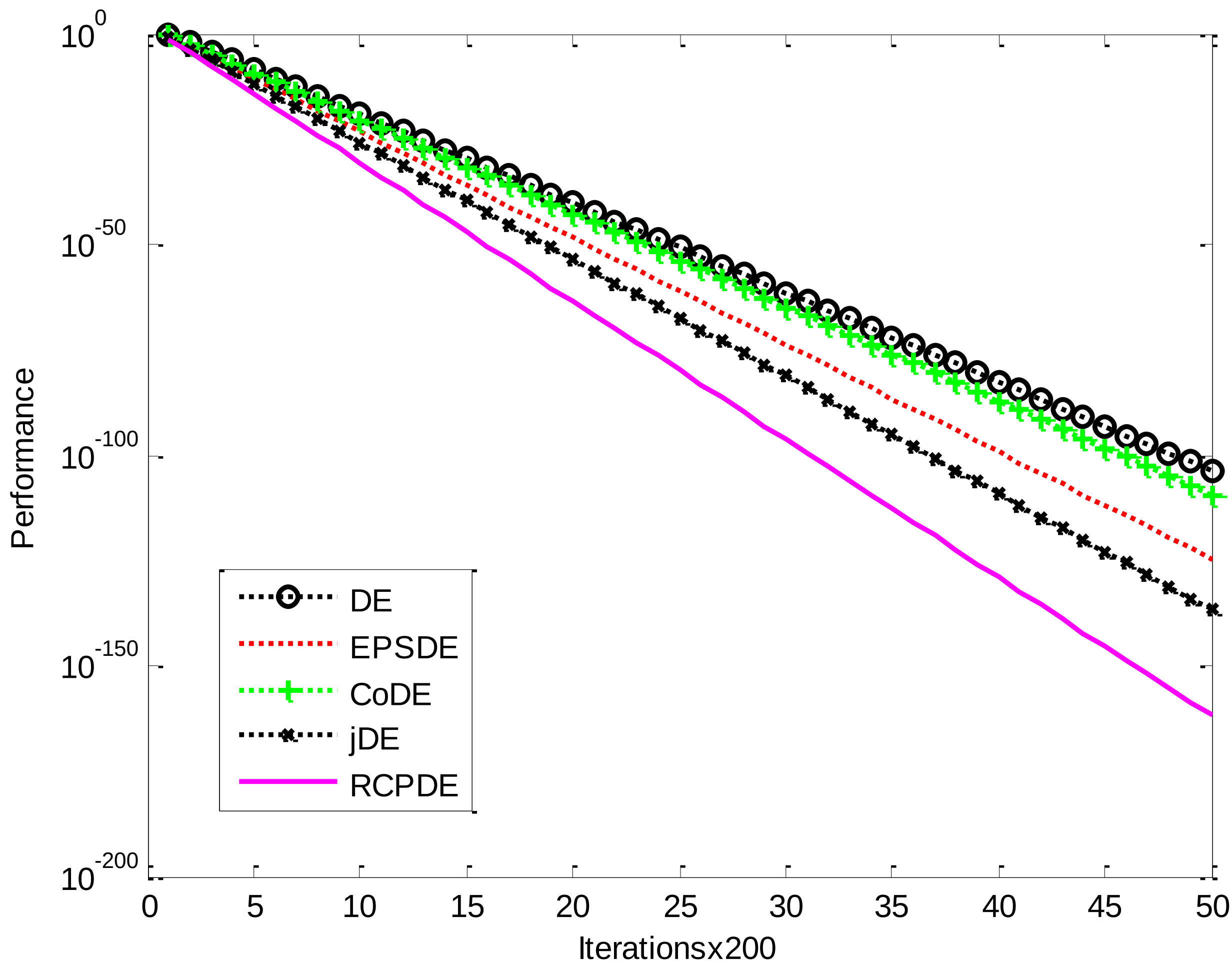


Figure 6.8- 20D average fitness logarithmic convergence graphs for f_{11} showing number of iterations horizontally and average fitness vertically

Figure 6.8 demonstrated the average fitness performance of RCPDE and other algorithms for Schwefel's problem-2.22 (f_{11}). This figure shows that the performance of DE algorithm and CoDE algorithm is similar in early iterations; however, CoDE has better performance in the succeeding iterations. It is clear from this figure that the performance of RCPDE is similar to other algorithms in early iterations, however in subsequent iterations its average fitness value decreases more quickly than other algorithms.

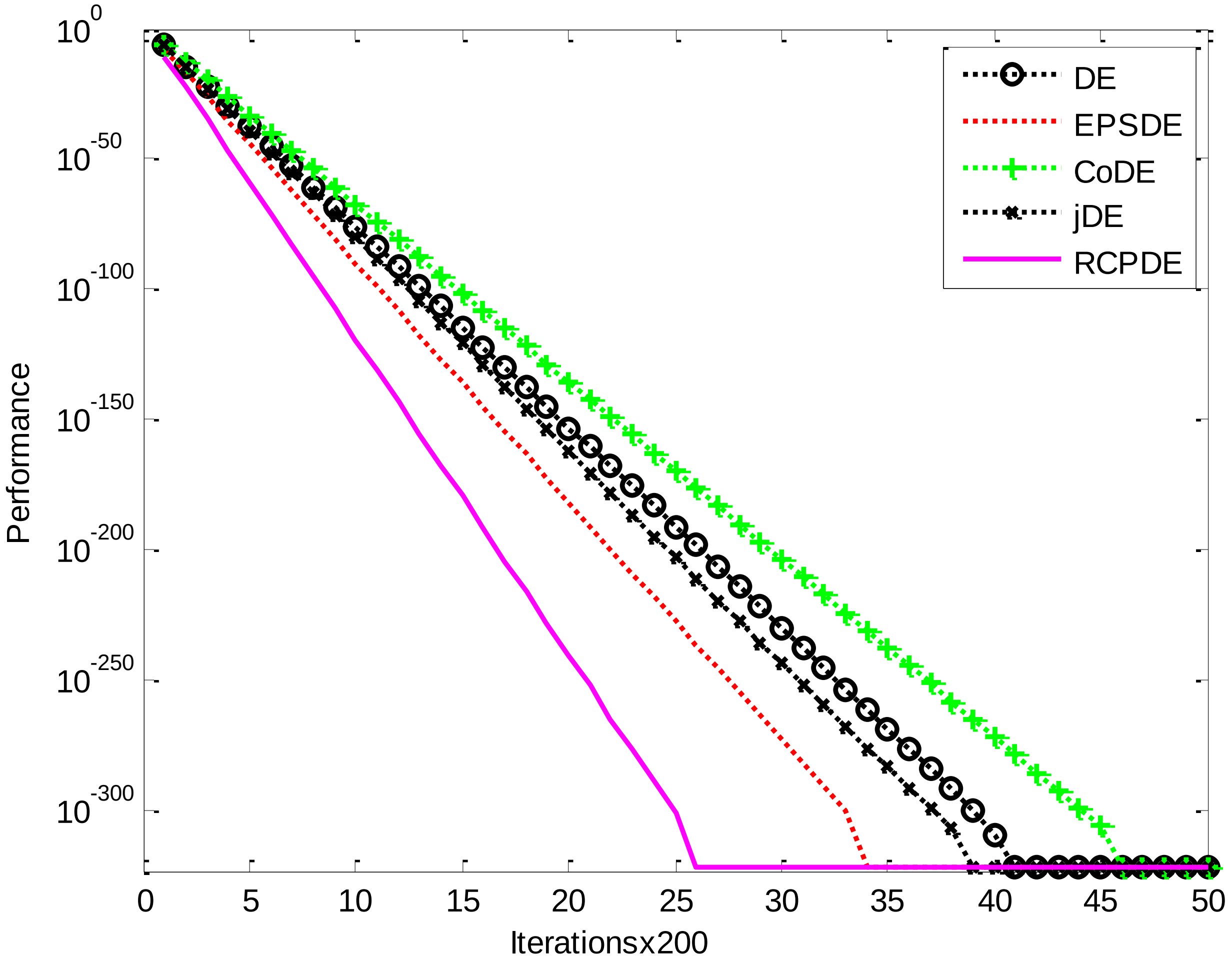


Figure 6.9- 20D average fitness logarithmic convergence graphs for f_{13} showing number of iterations horizontally and average fitness vertically

Figure 6.9 demonstrates the average fitness convergence performance graph of RCPDE and other algorithms for De Jong's function-4 (f_{13}). This figure illustrates that RCPDE, EPSDE, jDE, DE ad CoDE reached at optimal value within the specified iterations. RCPDE has quick convergence among all algorithm as it reaches at optimal value in small iterations while EPSDE, jDE, DE and CoDE achieves optimal value slowly than RCPDE as shown in figure. The important point to note is that the performance of RCPDE remains best from staring iterations to reach at optimal value.

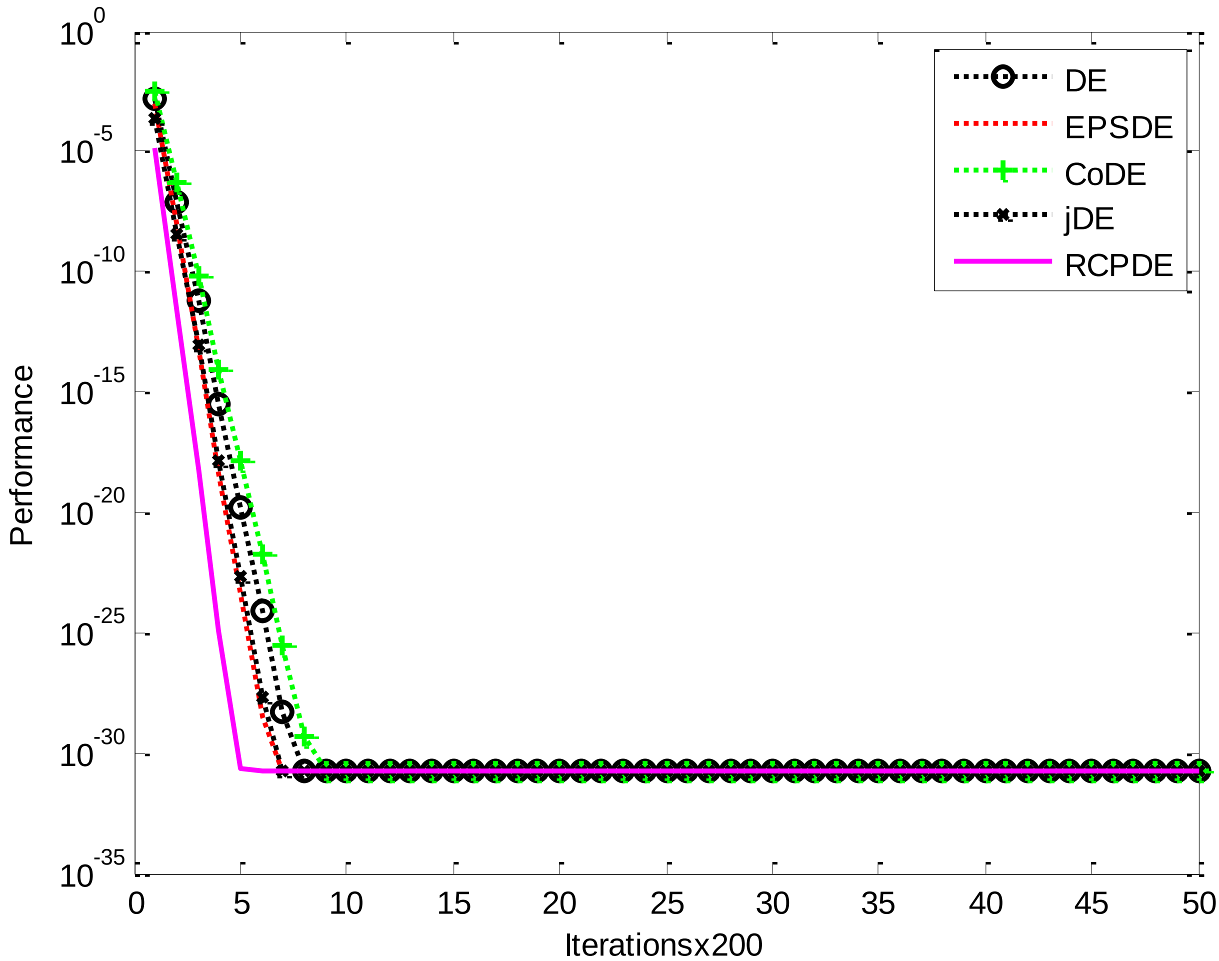


Figure 6.10- 20D average fitness logarithmic convergence graphs for f_{15} showing number of iterations horizontally and average fitness vertically

Figure 6.10 demonstrated average fitness performance of RCPDE and other algorithms for Levy and Montalvo Problem (f_{15}). This figure indicates that all algorithms reaches at value 3.27E-31 within specified iterations; however RCPDE is quick among these algorithms since it achieves this value in less iterations; however, DE, EPSDE, CODE and jDE achieves 3.27E-31 slower than RCPDE respectively. It is crucial to mention that RCPDE decreases more quickly than other algorithms from earlier iterations.

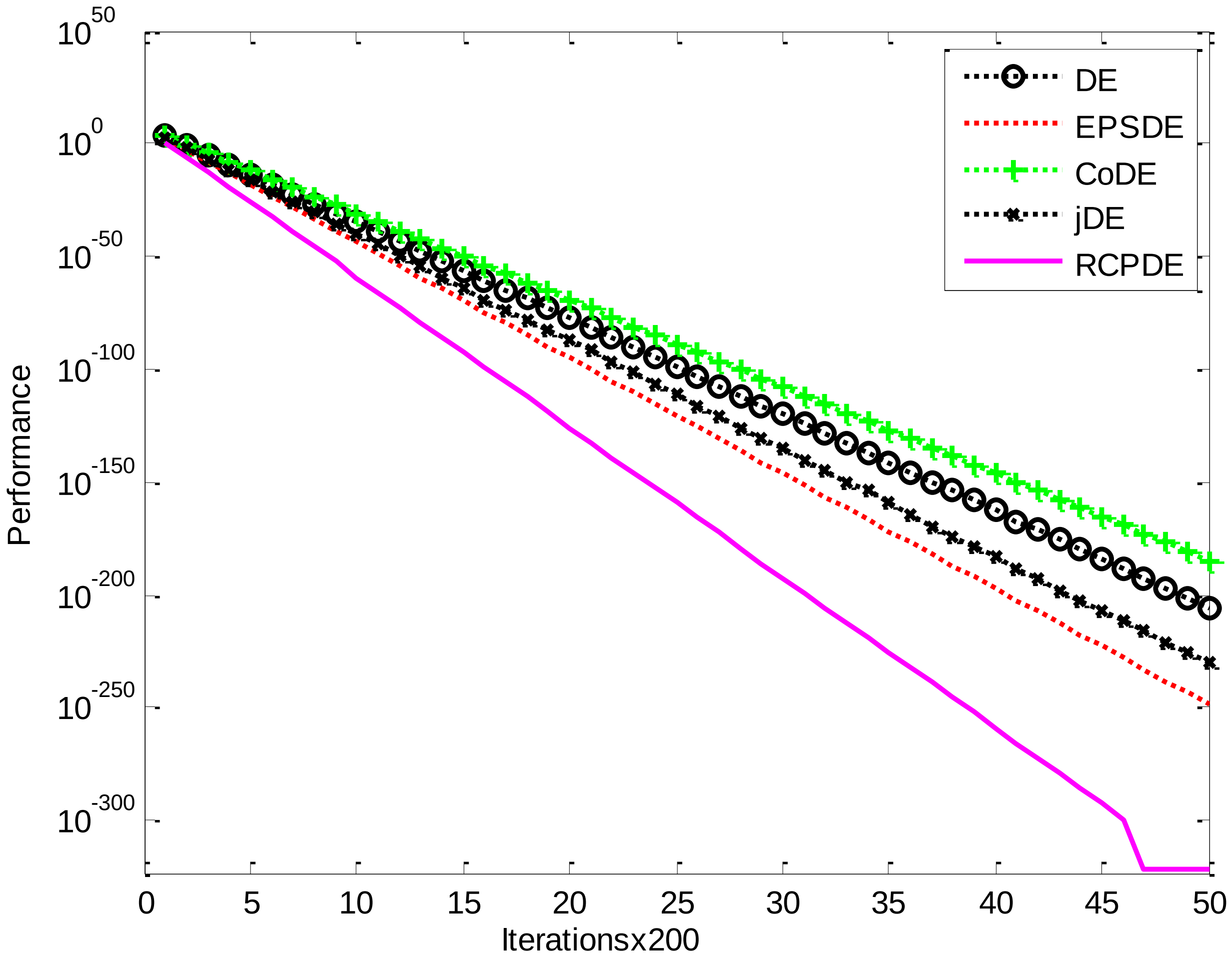


Figure 6.11- 20D average fitness logarithmic convergence graphs for f_{18} showing number of iterations horizontally and average fitness vertically

Figure 6.11 shows the average fitness of RCPDE & other algorithms for Cigar function (f_{18}). This figure depicts that RCPDE has fastest convergence among all other algorithms. It is observed from this figure that only RCPDE achieves optimum value 0 within the specified iterations. DE, jDE, EPSDE and CoDE have similar performance in the early iterations, however EPSDE has quick convergence and CoDE has slow convergence in the subsequent iterations. It is important to note that the convergence performance of RCPDE is better from initial iterations till final iteration.

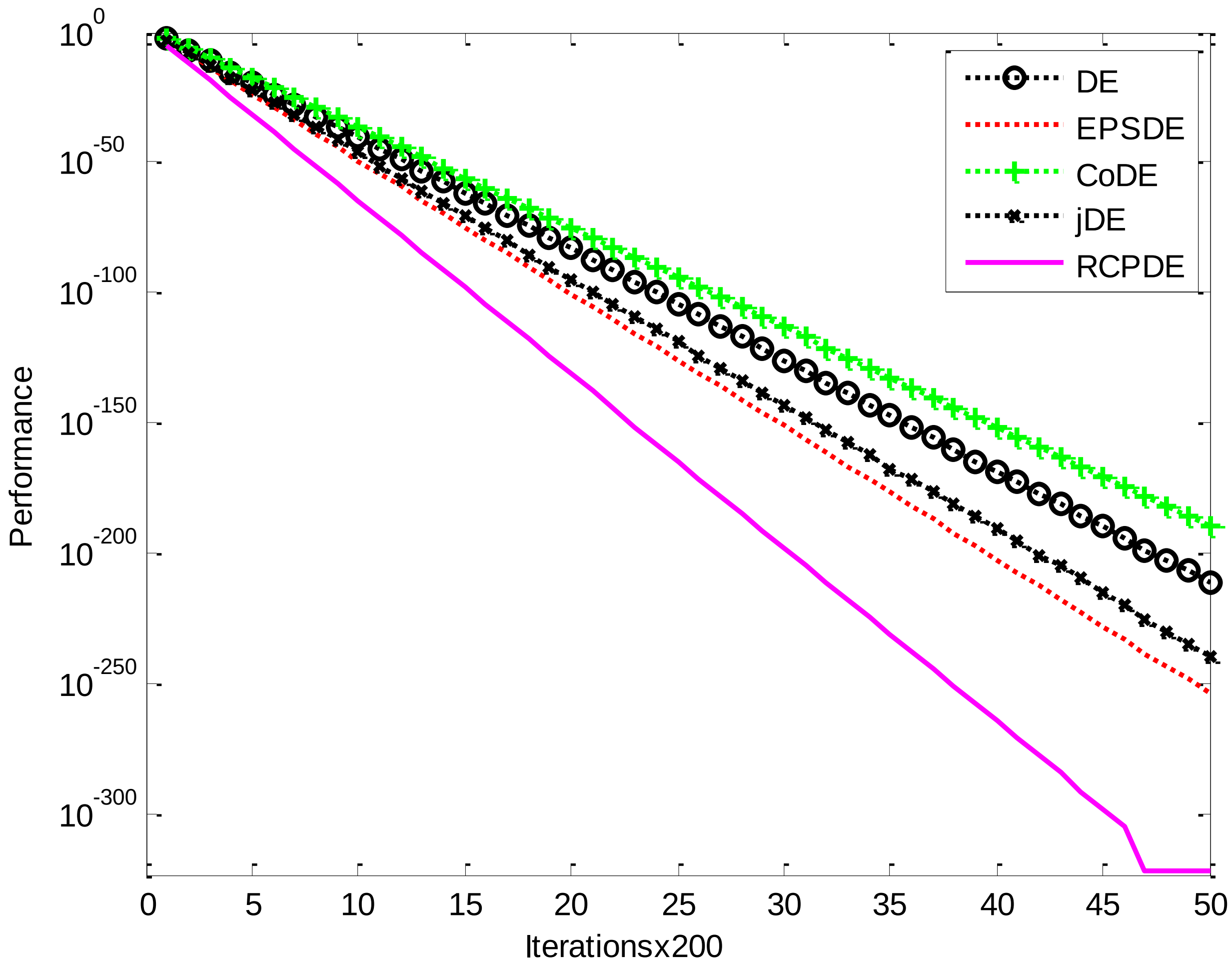


Figure 6.12- 20D average fitness logarithmic convergence graphs for f_{19} showing number of iterations horizontally and average fitness vertically

Figure 6.12 graphically presents the average fitness value of RCPDE and other algorithms for Function-15 (f_{19}) optimization problem. This figure indicates that CoDE has the worst performance among all algorithms till final iteration. The performance of most of algorithms in similar in the starting iterations, however, it varies in the subsequent iterations. It is important to mention that average fitness performance of RCPDE decreases more quickly from initial iteration to final iteration and it reaches at optimal value within the specified iterations.

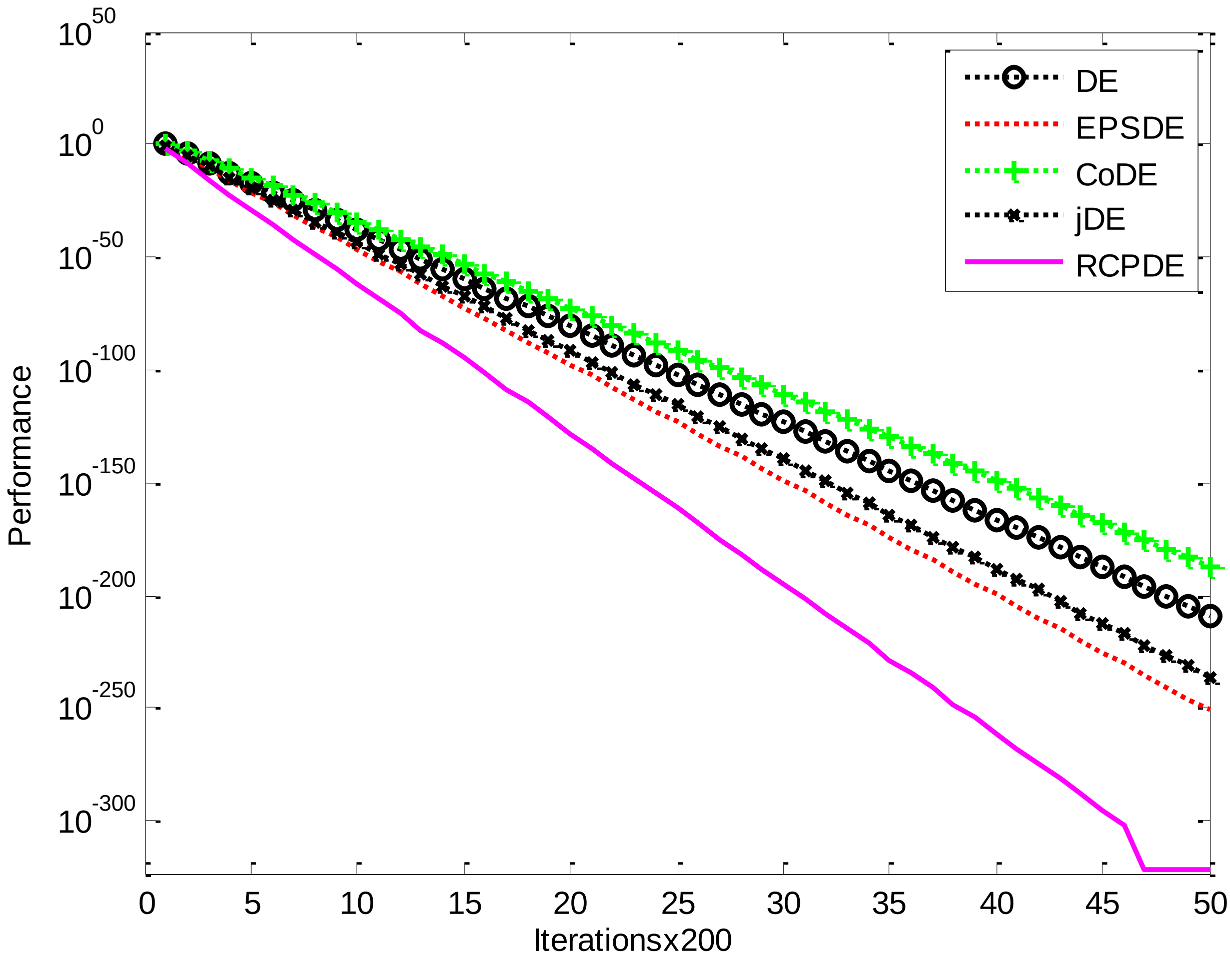


Figure 6.13- 20D average fitness logarithmic convergence graphs for f_{20} showing number of iterations horizontally and average fitness vertically

Average fitness performance of RCPDE and other algorithms is graphically presented in Figure 6.13 for Ellipse-Function (f_{20}). This figure illustrates that the performance of DE, EPSDE, CoDE and jDE is similar in the early iterations while it differs in the subsequent iterations. It is clear from this figure that the convergence performance of CoDE is worst among all algorithms for this problem. It is important to mention that RCPDE reaches at optimal value and maintains quick convergence speed among all competing algorithms.

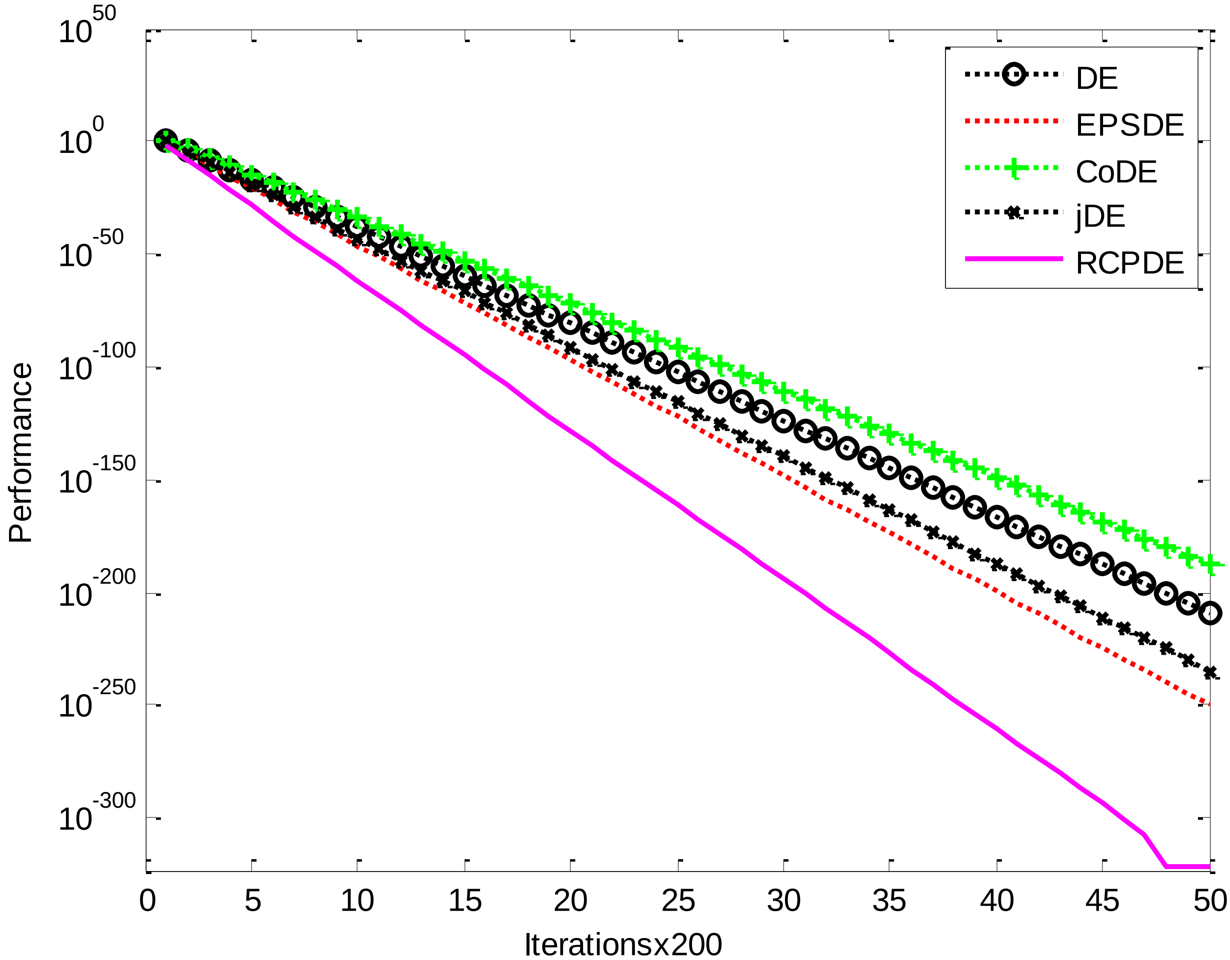


Figure 6.14- 20D average fitness logarithmic convergence graphs for f_{21} showing number of iterations horizontally and average fitness vertically

Figure 6.14 graphically presents an average fitness performance of RCPDE and other algorithms for Tablet Function (f_{21}). This figure shows that CoDE algorithm has worst convergence performance among all algorithms. It is obvious from this figure that RCPDE performance converges to optimum value within specified iterations. DE, EPSDE, CoDE and jDE have similar performance in starting iterations, however, EPSDE has quick convergence speed in the subsequent iterations and secures second best algorithm among all algorithms for this problem.

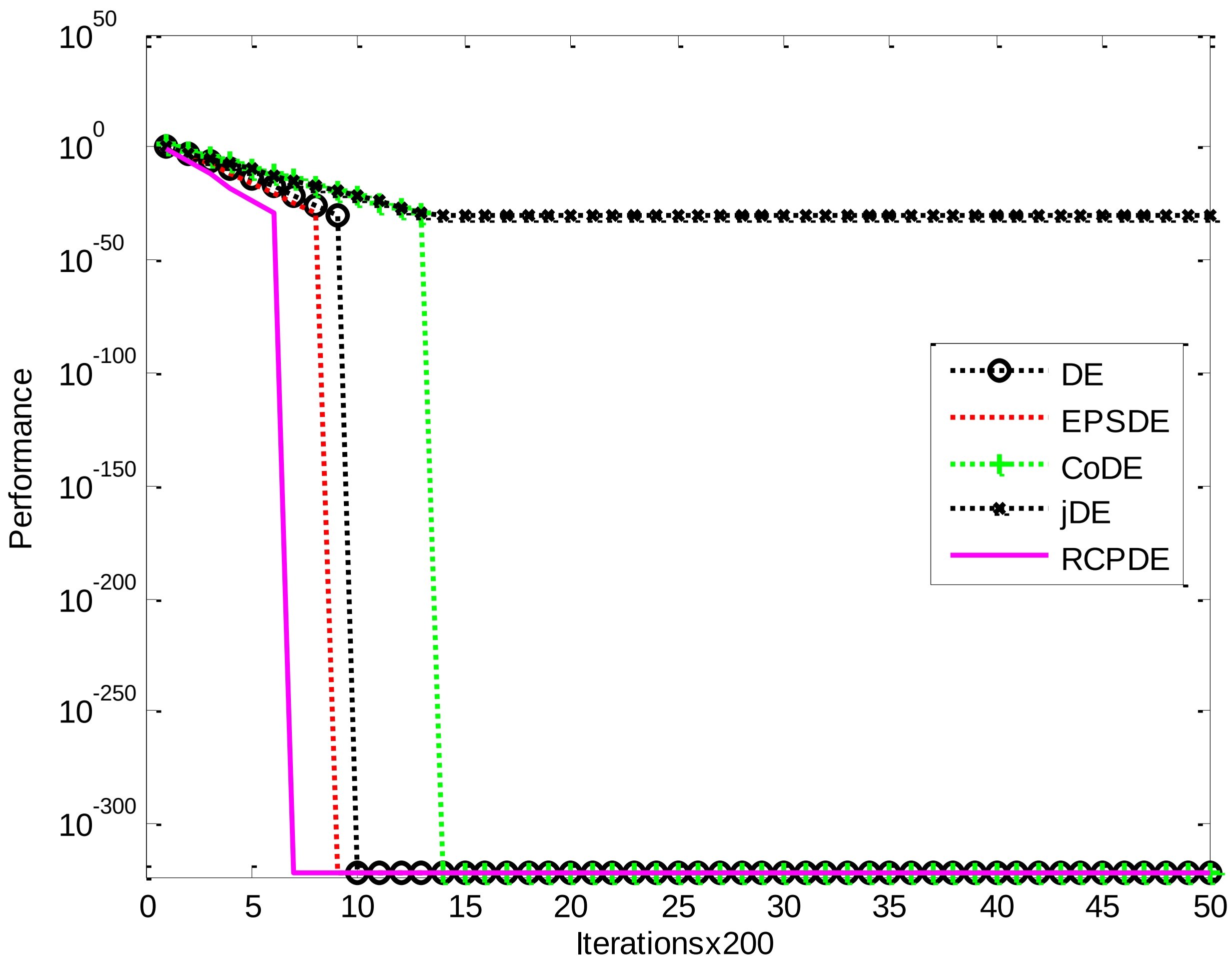


Figure 6.15 20D average fitness logarithmic convergence graphs for f_{22} showing number of iterations horizontally and average fitness vertically

Average fitness performance of RCPDE and other algorithms is graphically presented in Figure 6.15 for Schewel (f_{22}) problem. This figure depicts that the DE algorithm has the worst performance among other algorithms that achieves value 2.26E-03 in the final iteration. It is obvious from this figure that RCPDE reaches at optimum vale 0 quickly than EPSDE, CoDE, jDE and RCPDE. RCPDE reaches at optimal value in 700 iterations, while EPSDE, CoDE, jDE reaches at value 0 in 910, 1020 & 1390 iterations respectively.

It can be summarized from the simulation results and convergence graphs that the results of RCPDE are better than DE, EPSDE, CoDE and jDE. The inclusion of powerful control parameters in the parameter pool and diverse nature mutation strategies in strategy pool makes RCPDE algorithm capable to increase the convergence speed and solution quality as confirmed from results.

6.9 Summary

This chapter introduces RCPDE algorithm that uses a pool of mutation strategies and pool of control parameters. In RCPDE, mutation strategy pool and control parameter pool are helpful in balancing the exploration and exploitation ability of DE algorithm by incorporating potential mutation strategies and diverse control parameter values in DE algorithm. The two commonly used performance metrics NFC and Average fitness values are used to compare the performance of proposed DE variant with other DE state of the art algorithms. Convergence graphs of proposed RCPDE and other algorithms are given in Figure 6.3-15. Simulation results are presented in tables 6.1-4. A comprehensive set of N-dimensional benchmark functions is used to evaluate the performance of proposed improvement. Research result shows that RCPDE has significant performance.

Chapter # 7: Artificial Neural Network Training using TSDE and RCPDE for Data Classification

7.1 Chapter Summary

This chapter presents training of ANN using proposed TSDE and RCPDE algorithms. The results are also compared with other well known training ANN algorithms like BP algorithm, BP with Momentum algorithm, PSONN Algorithm.

7.1 Introduction

Many researchers have used DE algorithm to train artificial neural network. Hamed et. al have evolved multi-objective K-means evolving spiking neural network using DE algorithm(MO-KESNN) for clustering problem [190]. They have used standard UCI machine learning datasets to evaluate the performance of MO-KESNN. They have used two performance parameters number of pre-synaptic neurons measure and accuracy of clustering to show the significance of proposed model. Ibrahim et al. [191] have evolved multilayer perceptron architecture trained Backpropagation neural network algorithm using DE algorithm for breast cancer classification application. Their research result shows that the proposed solution is viable for the diagnoses of breast cancer. Chen and Song have evolved digital signal processor based neural network adaptive DE algorithm for thrust active magnetic bearing learned application [159]. The neural network trained using DE algorithm is capable of escaping from local optima problem and shows significant performance. Lin et al. [157] have used fuzzy DE (FDE) algorithm to evolve interactively recurrent functional neural fuzzy network (IRFNFN). IRFNFN reduces the number of nodes and focuses on information sharing of nodes in the interactive layer. The application of prediction and water bath temperature are used in their research work.

This chapter presents training of ANN for proposed TSDE and RPCDE algorithms. The training of ANN using both TSDE and RCPDE algorithms is validated by comparing the training results with the standard DE algorithm; commonly used DE mutation strategies and state-of-the-art versions are also obtained. For this purpose, training error and testing accuracy are used to assess the performance of TSDE and RCPDE.

7.2 Data sets and experimental setting

The data sets that are used to train ANN using DE algorithm and the experimental setting used to generate results are discussed in this section. The detail of datasets and parameter setting are discussed in the following subsection.

7.2.1 Datasets

To access the performance of proposed TSDE/bin and RCPDE algorithm, a test suit of standard benchmark datasets is used. The datasets are commonly used datasets taken from machine learning repository of datasets https: //archive.ics.uci.edu/ml/datasets.html

Table.7.1 Artificial Neural network classification datasets

Dataset	Attributes	# of instances	# of classes
Iris	4	150	3
Balance Scale	4	625	2
Breast Cancer	10	699	2
Haberman	3	306	2
Pima Indian	8	768	2
Seeds	7	210	3
Vertebral 2-Column	6	310	2
Heart	13	270	2
Fertility	10	100	2
Bank Note Authentication	4	1372	3
Diabetes	10	580	2
Mammographic	5	961	2
Customer	6	440	2
Wine	13	178	3

7.2.2 Experimental Setting

The experimental setting to train ANN using TSDE and RCPDE used in this thesis are given in table 7.2, 7.3 and 7.4.

Table.7.2 TSDE parameters for ANN Training

Population Size	30
Mutation Rate (F)	0.5
Crossover probability (CR)	0.7
Mutation strategy	"DE/rand/1/bin"(V_1), "DE/best/1/bin"(V_2), "DE/rand/2/bin"(V_3), "DE/best/2/bin"(V_4), "best/1/bin"(V_7), "DE/and repeated to best/1/bin"(V_{10}), "TSDE/bin" (V_{41}),

Table.7.3 RCPDE parameters for ANN Training

Population Size	30
Mutation Rate (F) and Crossover probability (CR) Pool	[F=1.0, Cr=0.1], [F=0.8, Cr=0.2], [F=0.5, Cr=0.9], [F=0.7, Cr=0.5]
Mutation strategies pool	"rand/1/bin", "rand/2/bin", "current-to-rand/1", "rand to best /1/bin", "TSDE/bin"

The ANN parameters used in this thesis are given in table 7.4.

Table.7.4 ANN parameter setting

Learning rate	0.5
Number of hidden layers	1
Number of hidden nodes in hidden layer	12
Training iterations	500
Architecture	Feed Forward

7.3 Evaluation and Validation

The ANN is trained using data classification applications. To evaluate the performance of TSDENN and RCPDENN two commonly used performance parameters are used.

- Training Error (MSE)
- Accuracy

7.4 ANN Training and testing

In ANN training features from datasets are extracted using any training algorithm. Various evolutionary and gradient based algorithms are used to train ANN algorithms. 85% of data is used as training data while15% data is used as testing data. K cross-fold validation method is used to assess the results of ANN training and testing. The value of K used is taken as 10.

7.5 Pseudocode of TSDENN Training

Figure 7.3 shows the pseudocode of ANN training using tournament selection base mutation variant of DE algorithm. The pseudocode contains an implementation flow of ANN training using TSDE algorithm that starts with the random initialization of ANN architectures. After initialization the fitness value/Mean Square Error (MSE) of each ANN architecture is calculated then ANN architectures are evolved by applying the DE algorithm using mutation, crossover and selection operators. The tournament selection criteria is used in the selection of parent architectures for proposed mutation strategy. After evolutionary process the optimal ANN architecture is obtained. TSDENN is also implemented through computer simulation.

Set BPNN parameters:

- *Maximum training iterations*
- *Learning rate*
- *Hidden neurons*
- *Neurons in input later*
- *Neurons in output layer*

1. Generate the initial ANN architectures $\mathrm{P}_{\mathrm{G}} = \{\mathrm{X}_{1,\mathrm{G}}, \ldots\ldots \mathrm{X}_{\mathrm{NP,G}}\}$ for generation *G*=0 and randomly initialize each ANN architectures $\mathrm{X}_{\mathrm{i,G}} = \{\mathrm{x}^{1}_{i,G} .... \mathrm{x}^{D}_{i,G}\}$ where *i* = 1...........NP(number of ANN architectures)

2. FOR *i* = 1 to NP

    Calculate mse $f(X_{i,G})$ for each ANN feed forward architectures $\mathrm{X}_{\mathrm{i,G}}$

   END FOR

3. WHILE the stopping criterion is not true

    / Start of TSDE vectors selection */*

   Step 3.1 TSDE vectors selection

    FOR *n* = 1 to number of *TSDE* ANN architectures

     FOR *k* = 1 to *Tournment_size*

      Select randomly k^{th} tournament ANN architectures along with its mse from current population

     END FOR

     Select best of best ANN architectures from the current tournament as $\boldsymbol{n}^{th}$ *tbest*

      Return $\boldsymbol{n}^{th}$ ANN architectures index to be used as one of *TSDE* ANN architecture in proposed mutation strategy

    END FOR

    / End of TSDE vectors selection */*

   Step 3.2 Mutation Step

     FOR *i* = 1 to NP

For the i^th^ target ANN architecture $X_{i,G}$ generate a donor vector $V_{i,G} = \left\{ v_{i,G}^{1} .... v_{i,G}^{D} \right\}$ with the specified mutation strategy (Given in Table-7.1)

END FOR

Step 3.3 Crossover Step

FOR *i* = 1 to NP

For the i^th^ target ANN architecture $X_{i,G}$ generate a trial ANN architecture $U_{i,G} = \left\{ u_{i,G}^{1} .... u_{i,G}^{D} \right\}$ with the specified crossover scheme (given in section 2.5)

END FOR

Step 3.4 Selection Step

FOR *i*=1 to NP

Evaluate the trial ANN feed forward architecture $U_{i,G}$ against the target ANN feed forward architecture $X_{i,G}$ with mse function f

IF $f(U_{i,G} \le f\left(X_{i,G}\right)$, THEN $X_{i,G+1} = U_{i,G}$, $f(X_{i,G}) = f\left(U_{i,G}\right)$

IF $f(U_{i,G} \le f\left(X_{best,G}\right)$, THEN $X_{best,G+1} = U_{i,G}$,

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

END IF

END IF

END FOR

Step 3.5 Update mse

Step 3.6 increment generation number G=G+1

END WHILE

4. *Get Optimum ANN architecture*

Figure 7.1 *Pseudocode of ANN training algorithm using Tournament Selection based DE algorithm (TSDENN)*

7.6 Flowchart of TSDE-ANN Training

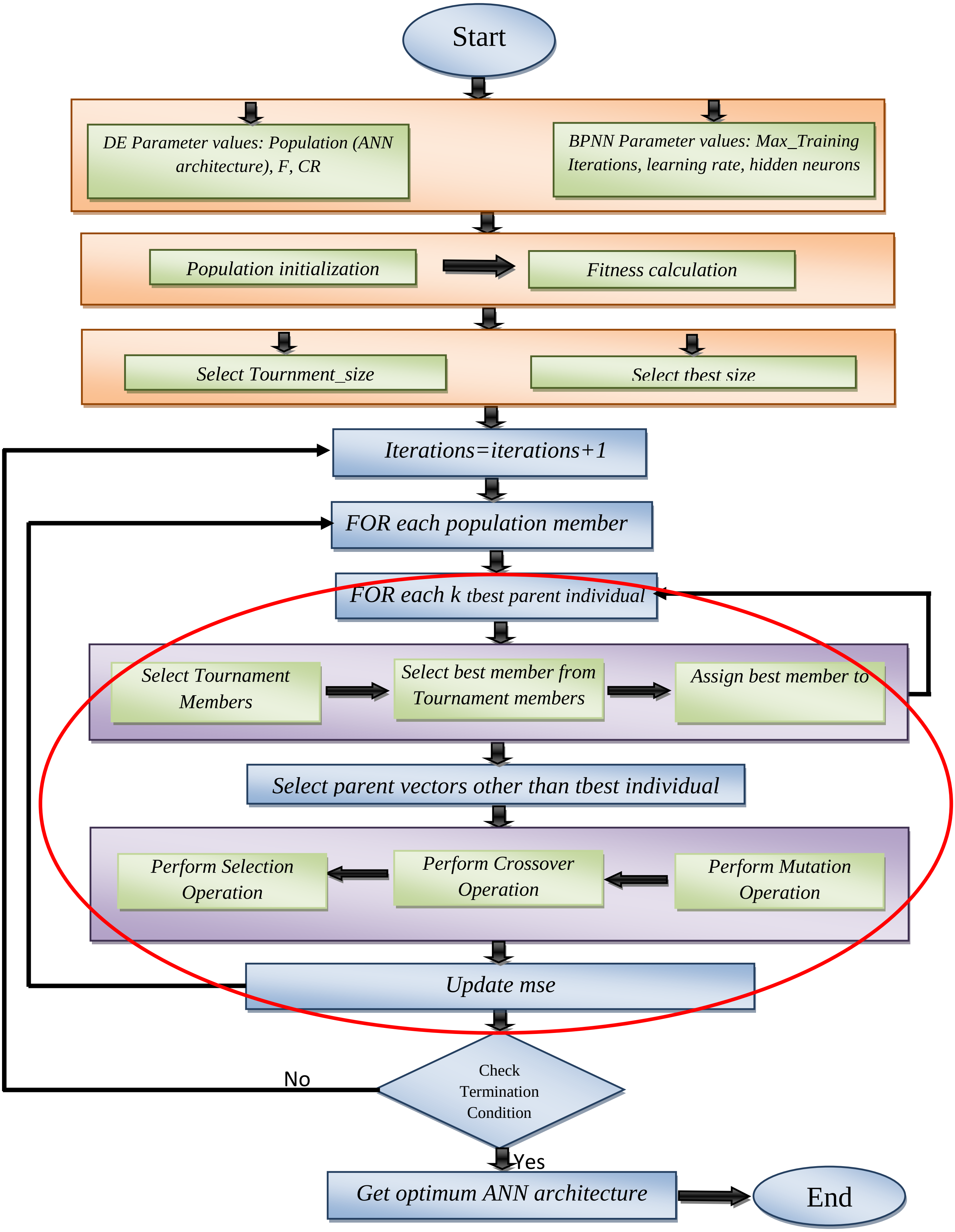


Figure 7.2 *Flowchart of ANN training algorithm using Tournament Selection based DE algorithm (TSDENN)*

7.7 Accuracy of results

Table 7.5 Accuracy of results (Mean ± Standard Deviation)

Data Set	V_1	V_2	V_3	V_4	V_7	V_{10}	V_{41}
Iris	92.67±7.98	**96.00±7.17**	86.67±16.02	**96.00±5.62**	95.33±8.92	95.33±5.49	95.33±8.34
Balance Scale	70.71±14.48	73.02±14.29	**75.71±5.15**	72.30±14.40	68.73±15.17	71.19±14.95	71.90±15.01
Breast Cancer	96.57±2.10	96.29±2.81	96.29±1.46	96.57±1.64	96.50±2.06	96.64±2.48	**96.64±1.96**
Haberman	72.33±3.78	73.00±1.05	73.17±2.00	73.00±0.70	**73.33±0.00**	**73.33±0.00**	**73.33±0.00**
Pima Indian	**75.52±3.14**	75.32±2.65	74.29±4.22	75.13±4.75	74.61±3.79	75.00±4.18	74.42±2.67
Seeds	87.86±6.78	88.57±4.87	82.14±11.51	89.76±5.94	89.52±4.92	89.52±6.17	**90.48±4.35**
Vertebral 2-Column	75.97±13.59	79.52±9.59	76.94±10.51	77.10±16.25	77.74±11.37	77.74±10.64	**78.87±9.17**
Heart	80.56±8.01	81.85±4.17	80.19±6.30	81.30±4.74	80.19±2.90	82.41±3.06	**82.78±3.91**
Fertility	84.00±6.58	83.50±10.55	87.00±2.58	86.00±7.75	85.00±9.13	**87.50±7.17**	82.22±13.25
Bank Note Authentication	96.53±1.09	98.32±0.43	95.44±1.94	96.97±0.73	97.66±0.98	98.28±1.03	**98.47±0.54**
Diabetes	67.50±4.81	67.50±5.25	67.67±2.97	66.81±6.56	69.74±4.55	**70.34±4.21**	68.97±6.81
Mammographic	**81.61±2.38**	81.56±2.82	80.05±4.39	80.31±2.93	80.31±3.53	80.57±3.38	80.89±3.45
Customer	**91.36±4.39**	90.11±4.22	90.00±5.22	89.55±4.90	89.89±4.43	90.00±4.84	90.45±4.68
Wine	86.94±7.41	93.33±5.27	81.67±8.61	88.33±6.52	93.89±4.50	93.89±3.88	**94.44±2.93**

The accuracy of result shows that the proposed mutation strategy V_{41}has overall best performance among all mutation strategies. In most of the cases the performance of V_{41} is either better or comparable to other mutation strategies. The results clearly indicated that V_{41} successful trains ANN and has significant performance.

7.8 Training Error (Mean ± Standard Deviation)

Table.7.6 Training Error (Mean ± Standard Deviation)

	V_1	V_2	V_3	V_4	V_7	V_{10}	V_{41}
Iris	1.59E-02±4.73E-03	6.86E-03±1.70E-03	2.25E-02±5.27E-03	1.22E-02±3.00E-03	7.44E-03±1.67E-03	9.04E-03±5.00E-03	6.91E-03±2.82E-03
Balance Scale	3.05E-02±4.73E-03	2.40E-02±1.70E-03	3.39E-02±5.27E-03	2.80E-02±3.00E-03	2.46E-02±1.67E-03	2.04E-02±5.00E-03	1.81E-02±2.03E-03
Breast Cancer	1.09E-02±1.83E-03	7.13E-03±2.03E-03	1.18E-02±1.84E-03	9.78E-03±2.15E-03	7.08E-03±2.14E-03	6.66E-03±2.04E-03	5.36E-03±1.69E-03
Haberman	6.62E-02±2.32E-02	5.72E-02±2.10E-02	6.91E-02±2.33E-02	6.25E-02±2.30E-02	5.82E-02±2.16E-02	4.86E-02±1.80E-02	4.30E-02±1.57E-02
Pima Indian	1.12E-01±2.92E-03	9.40E-02±3.30E-03	1.18E-01±3.04E-03	1.06E-01±4.64E-03	9.57E-02±3.91E-03	8.24E-02±2.48E-03	7.24E-02±2.40E-03
Seeds	2.71E-02±3.67E-03	9.85E-03±3.30E-03	3.22E-02±5.05E-03	2.12E-02±2.55E-03	1.23E-02±3.32E-03	1.21E-02±2.90E-03	4.30E-02±1.57E-02
Vertebral 2-Column	6.15E-02±7.55E-03	4.47E-02±1.01E-02	6.51E-02±1.08E-02	5.57E-02±8.94E-03	4.79E-02±8.53E-03	3.91E-02±7.03E-03	4.30E-02±1.57E-02
Heart	5.86E-02±5.83E-03	3.97E-02±4.02E-03	7.04E-02±4.99E-03	5.53E-02±5.34E-03	4.16E-02±4.90E-03	3.56E-02±5.19E-03	3.04E-02±4.00E-03
Fertility	3.38E-02±5.47E-03	1.85E-02±7.20E-03	3.63E-02±4.30E-03	3.02E-02±5.20E-03	1.94E-02±9.68E-03	1.88E-02±5.46E-03	1.34E-02±6.06E-03
Bank Note Authentication	1.04E-02±1.76E-03	2.28E-03±1.11E-03	1.26E-02±1.20E-03	8.23E-03±1.24E-03	3.34E-03±1.34E-03	2.25E-03±1.16E-03	2.00E-03±6.45E-04
Diabetes	1.33E-01±2.63E-03	1.20E-01±4.60E-03	1.34E-01±2.26E-03	1.29E-01±2.24E-03	1.24E-01±2.42E-03	1.06E-01±2.70E-03	9.36E-02±2.53E-03
Mammographic	2.32E-03±3.24E-03	2.98E-03±3.73E-03	2.32E-03±3.26E-03	2.36E-03±2.94E-03	2.92E-03±3.30E-03	2.16E-03±2.94E-03	5.83E-02±2.23E-03
Customer	3.61E-02±4.72E-03	2.39E-02±3.47E-03	3.84E-02±4.90E-03	3.22E-02±5.08E-03	2.49E-02±3.80E-03	2.07E-02±4.38E-03	1.83E-02±3.97E-03
Wine	1.80E-02±4.49E-03	3.10E-04±6.07E-04	2.88E-02±4.54E-03	7.34E-03±2.79E-03	1.38E-06±3.09E-06	8.58E-07±2.37E-06	1.98E-05±6.26E-05

The Mean Square Error results of mutation "DE/rand/1/bin"(V_1), "DE/best/1/bin"(V_2), "DE/rand/2/bin"(V_3), "DE/best/2/bin"(V_4), "best/1/bin"(V_7), "DE/and repeated to best/1/bin"(V_{10}), "TSDE/bin" (V_{41}) are shown in table 7.6 for some classification applications given in table 7.1. The detailed analysis and convergence of mean square error are discussed in the following convergence graphs given in figures 7.3 to 7.12.

7.9 Error Convergence Graphs

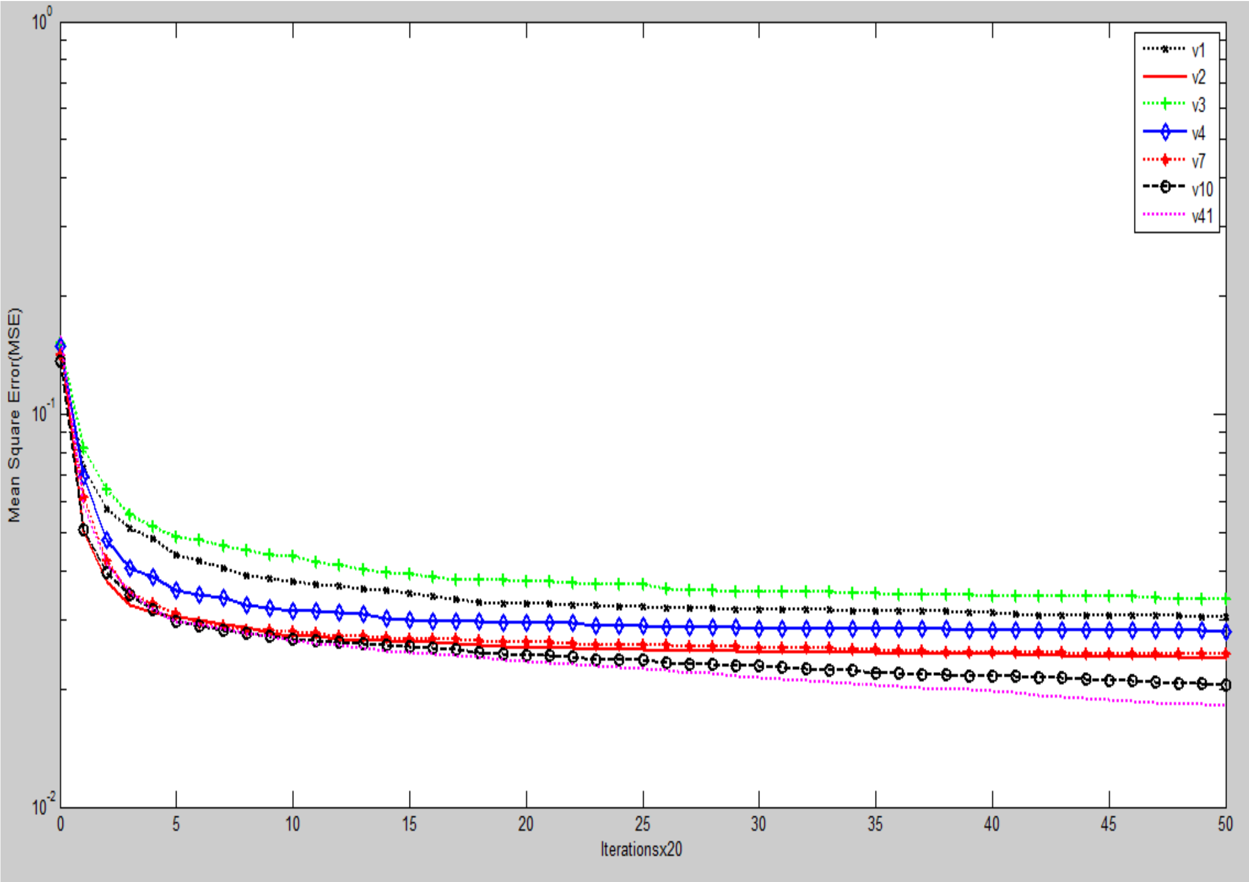


Fig.7.3 Balance Scale Training Error Convergence

Mean square achieved by the proposed *"TSDE/bin" (V41)* and other conventional techniques for Balance Scale data classification problem is graphically presented in Figure 7.3. This figure depicts that MSE of proposed *"TSDE/bin" (V41)* in the first iteration is 10^{-1} which is gradually decreasing in all subsequent iterations and reaches to 10^{-2} in the last iteration. The MSE value achieved by *"DE/best/1/bin"(V_2), "DE/rand/2/bin"(V_3), "DE/best/2/bin"(V_4)* and *"DE/current to best/1/bin"(V_7)* is $10^{-0.7}$, $10^{-0.6}$, $10^{-0.7}$, $10^{-0.7}$, $10^{-0.8}$ and $10^{-0.9}$ respectively in the last iteration. However, the MSE value of *"TSDE/bin" (V41)* in last iteration is $10^{-0.8}$ which is smaller than all other conventional techniques.

The performance of proposed *"TSDE/bin" (V41)* for classification of Breast Cancer data in term of MSE value is better than other conventional variants as shown in Figure 7.4. From the figure it is observed that *"DE/rand/2/bin"(V_3)* has worse performance, which achieves MSE value 10^{-1} in the last iteration. It is noted that the MSE value of *"DE/rand repeated to best/1/bin"(V_{10}), "DE/best/1/bin"(V_2)* and *"TSDE/bin" (V41)* is similar in early iterations;

however, *“TSDE/bin” (V41)* MSE decreases more quickly than *"DE/best/1/bin"*(V_2) and *"DE/rand repeated to best/1/bin"*(V_{10}) in the subsequent iterations.

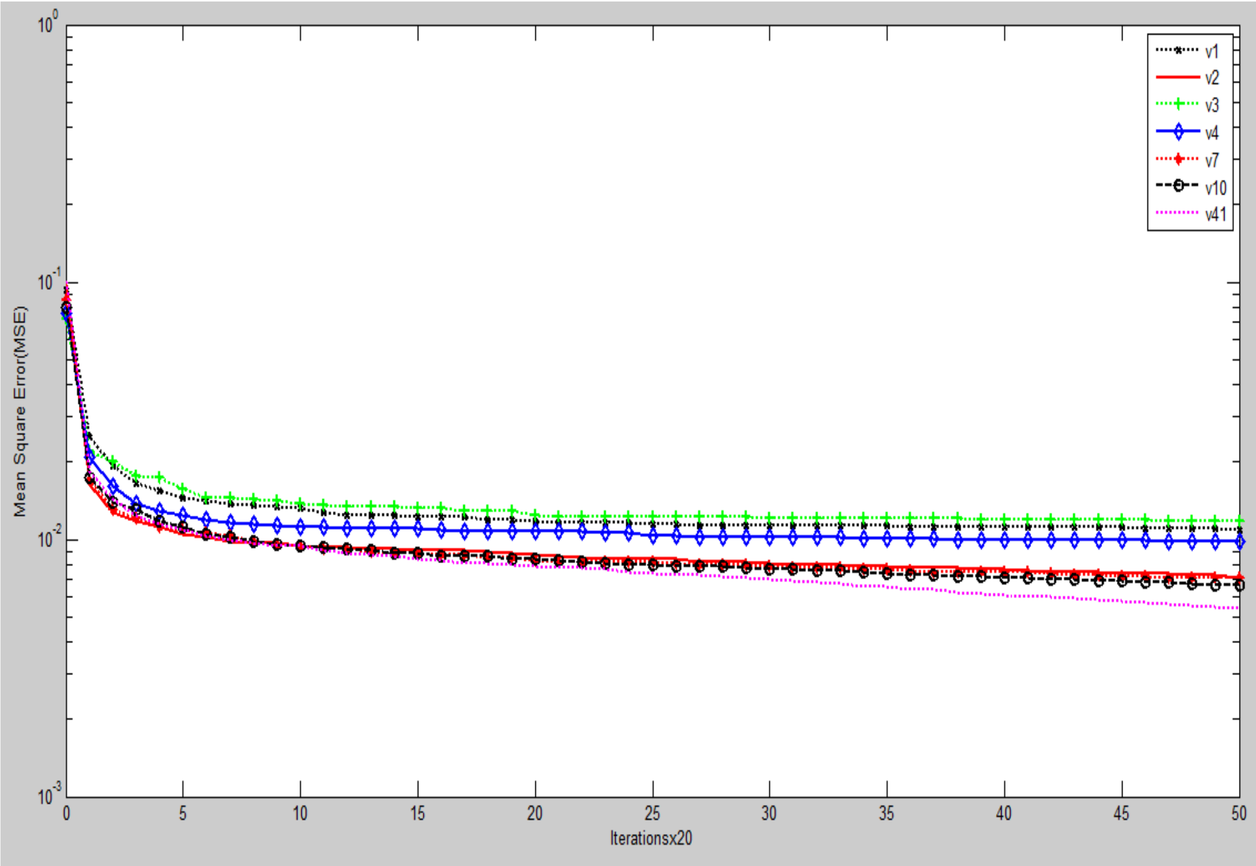


Fig.7.4 Breast Cancer Training Error Convergence

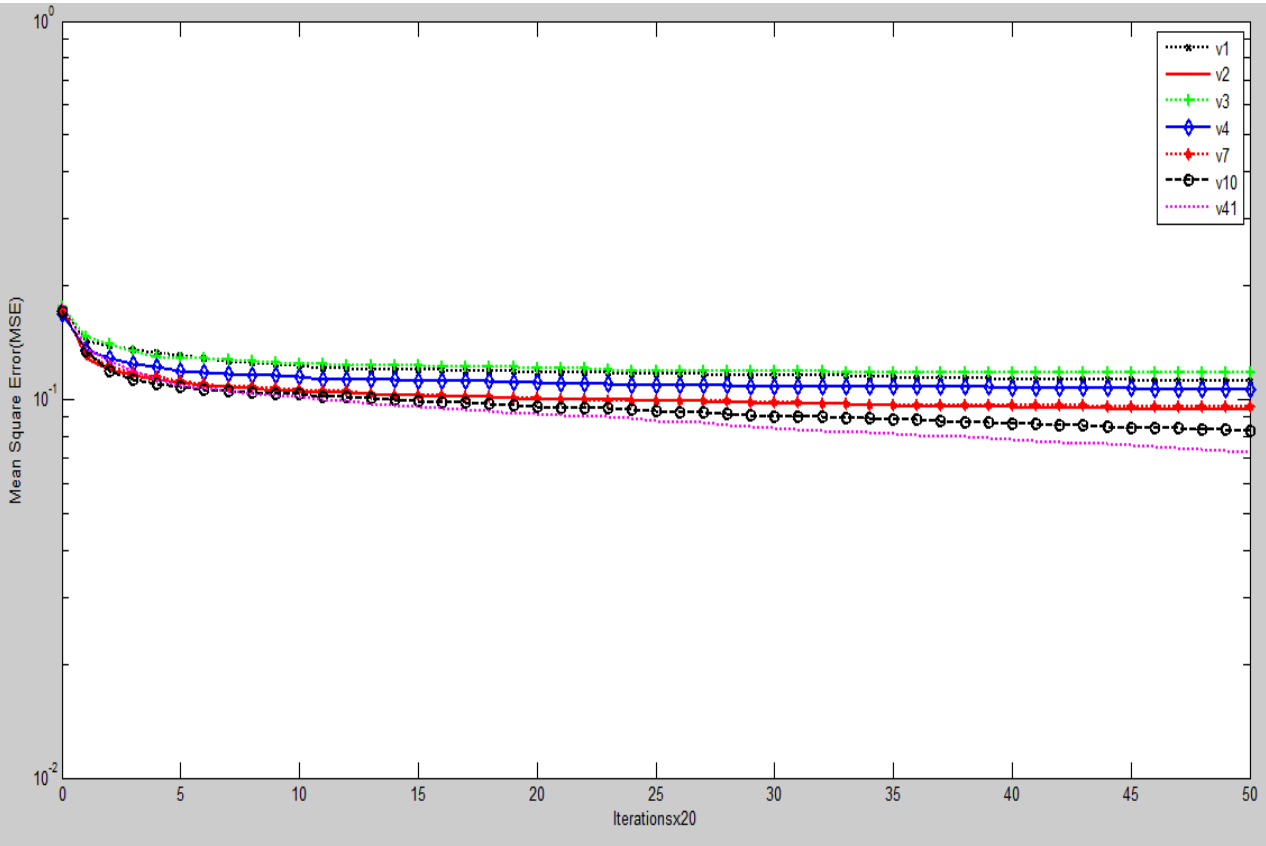


Fig.7.5 Pima Indian Training Error Convergence

Mean square error of proposed *“TSDE/bin” (V41)* & other mutation strategies is graphically presented in Figure 7.5 for Pima Indian classification problem. The mean square error of *“TSDE/bin” (V41)* is similar to *"DE/rand repeated to best/1/bin"(V_{10})* in first 100 iterations; however, it gradually decreases in the subsequent iterations. The mean square error achieved by *“TSDE/bin” (V41)* in final iterations is 10^{-2} which is smallest than all other mutation strategies. It is important to note that *"DE/rand/2/bin"(V_3)* has worst performance among all mutation strategies. Another important to mention is that *"DE/rand repeated to best/1/bin"(V_{10})* has second best convergence for this application.

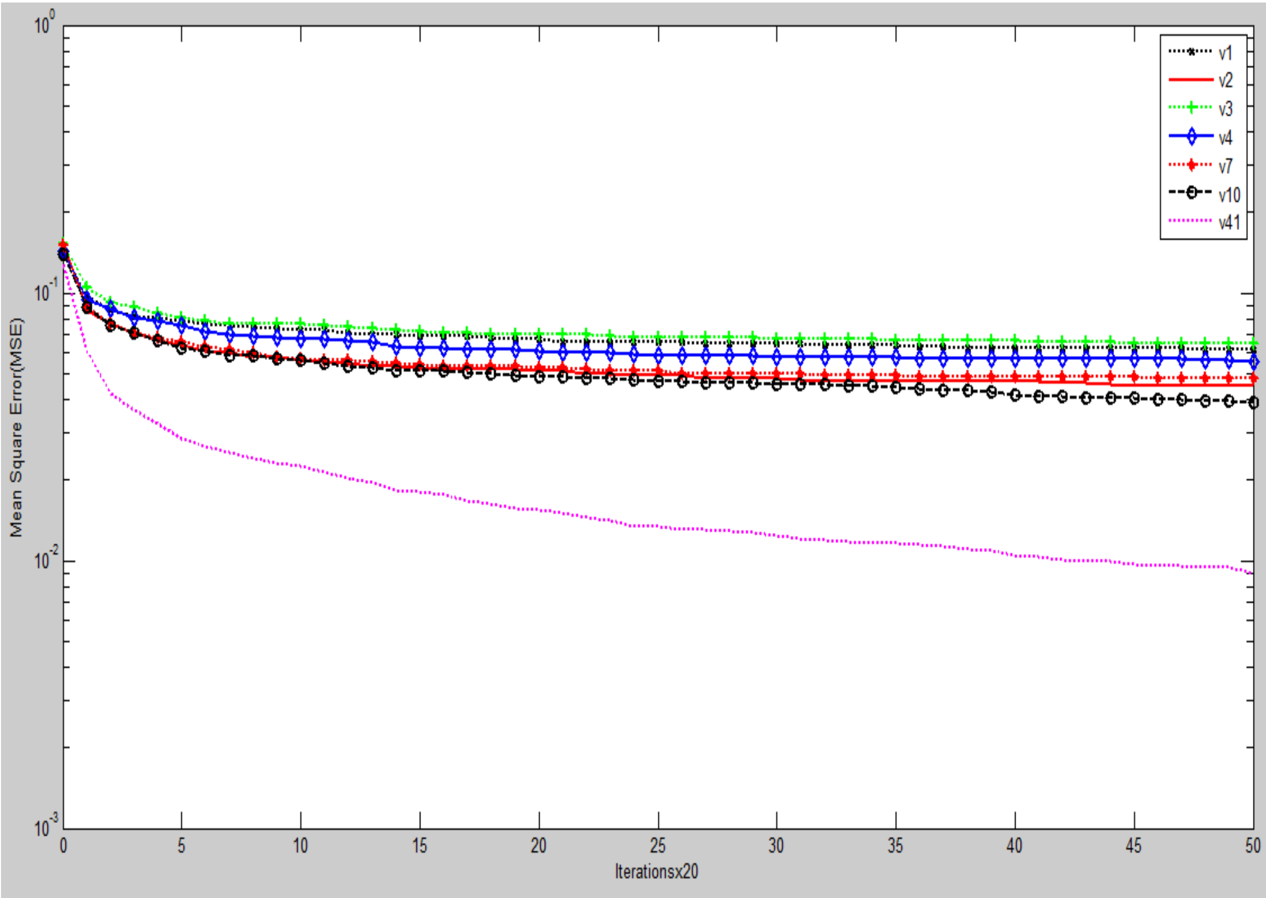


Fig.7.6 Seed Training Error Convergence

The mean square error for seed classification data is graphically presented in Figure 7.6. This figure depicts that the mean square of proposed *“TSDE/bin” (V41)* starts at $10^{-0.8}$ and continuously decreases to a value $10^{-2.1}$ in final iteration which is best among all mutation strategies. It is important to note that the mean square error of *“TSDE/bin” (V41)* decreases quickly than all other mutation strategies. Mutation strategy *"DE/rand/2/bin"(V_3)* has worst performance and *"DE/rand repeated to best/1/bin"(V_{10})* achieves second best convergence in the specified iterations.

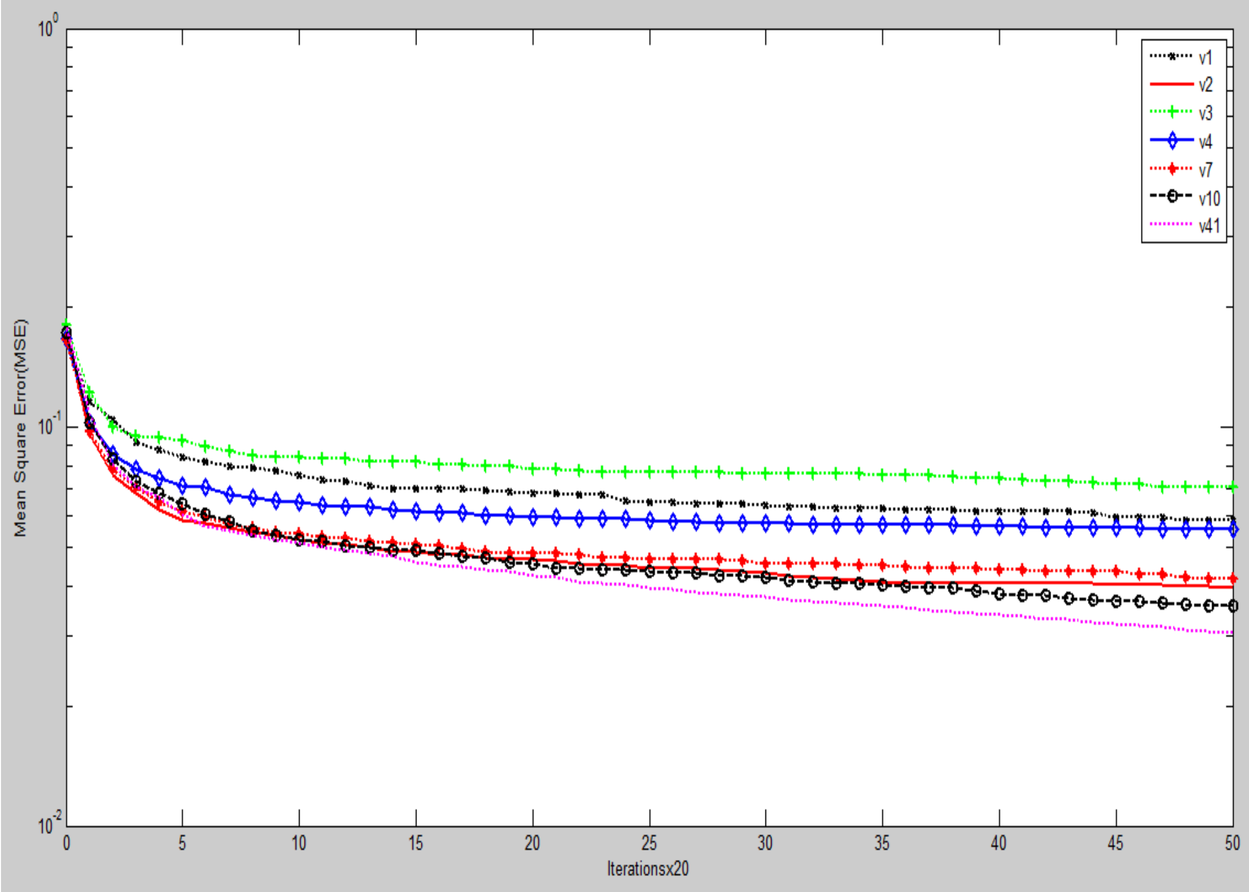


Fig.7.7 Heart Training Error Convergence

Mean square error achieved by proposed *"TSDE/bin" (V41)* & other conventional techniques for heart classification data is graphically presented in Figure 7.7. This figure illustrates that MSE of proposed *"TSDE/bin" (V41)* is similar to *"DE/rand repeated to best/1/bin"(V_{10})*, *"DE/best/1/bin"(V_2)* & *"DE/current to best/1/bin"(V_7)* till 210th iteration then it starts gradually decreasing till final iteration. Mutation strategy *"TSDE/bin" (V41)* achieves MSE $10^{-1.2}$ in final iteration, which is best among all other mutation strategies. It is important to mention that *"DE/rand repeated to best/1/bin"(V_{10})* has second best MSE than *"DE/rand/1/bin"(V_1)* ,*"DE/best/1/bin"(V_2)*, *"DE/rand/2/bin"(V_3)*, *"DE/best/2/bin"(V_4)* & *"DE/current to best/1/bin"(V_7)* which is 10^{-1} in the final iteration.

Figure 7.8 contains MSE of proposed *"TSDE/bin" (V41)* & other mutation strategies for fertility data classification problem. This figure depicts that MSE of *"TSDE/bin" (V41)* in the first 200 iterations is similar to *"DE/best/1/bin"(V_2)* and *"DE/current to best/1/bin"(V_7)*; however, it continuously decreases in the succeeding iterations. The MSE achieved by *"TSDE/bin" (V41)* in final iteration is $10^{-1.8}$ which is smallest among all mutation strategies. It is important to mention that *"DE/rand/2/bin"(V_3)* achieve worst performance than all other mutation strategies.

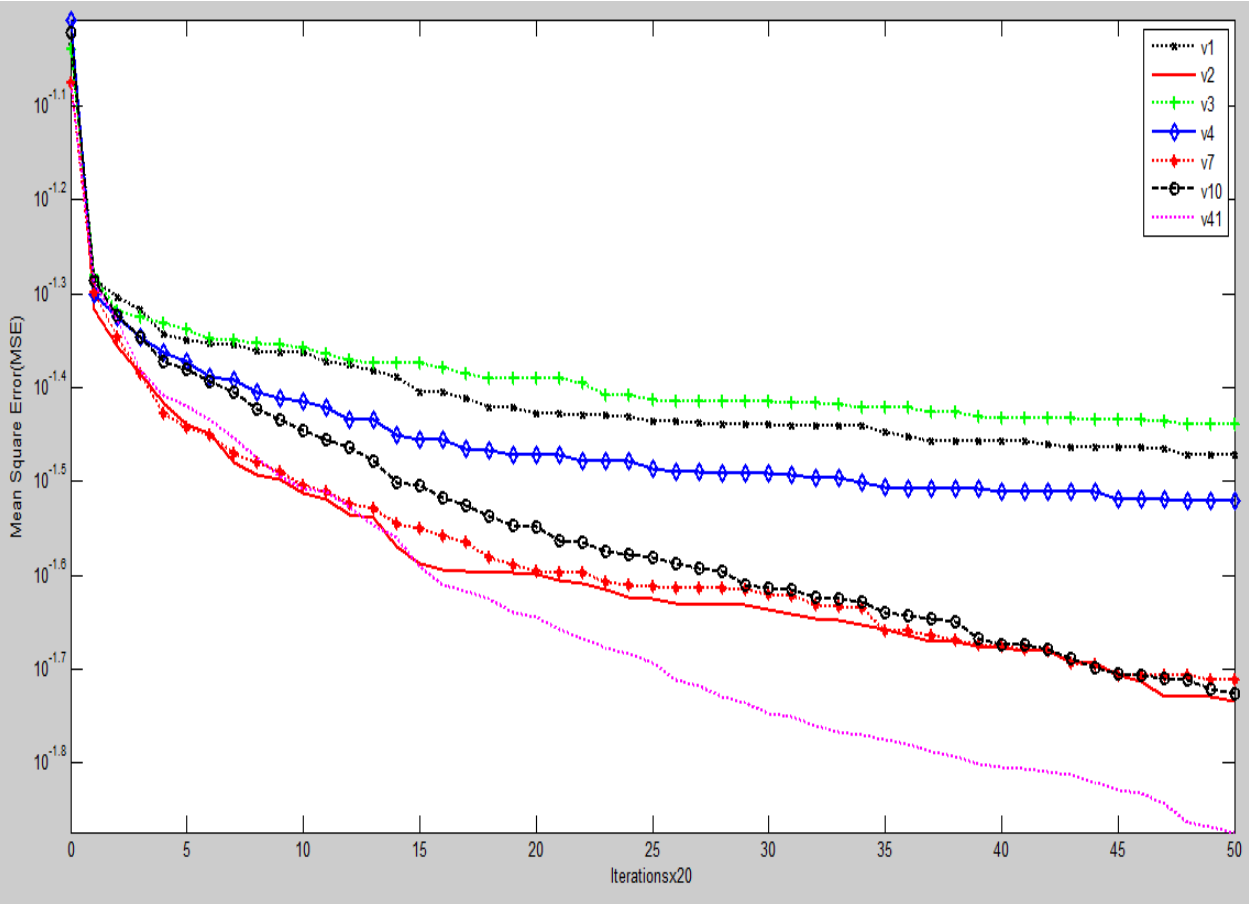


Fig.7.8 Fertility Training Error Convergence

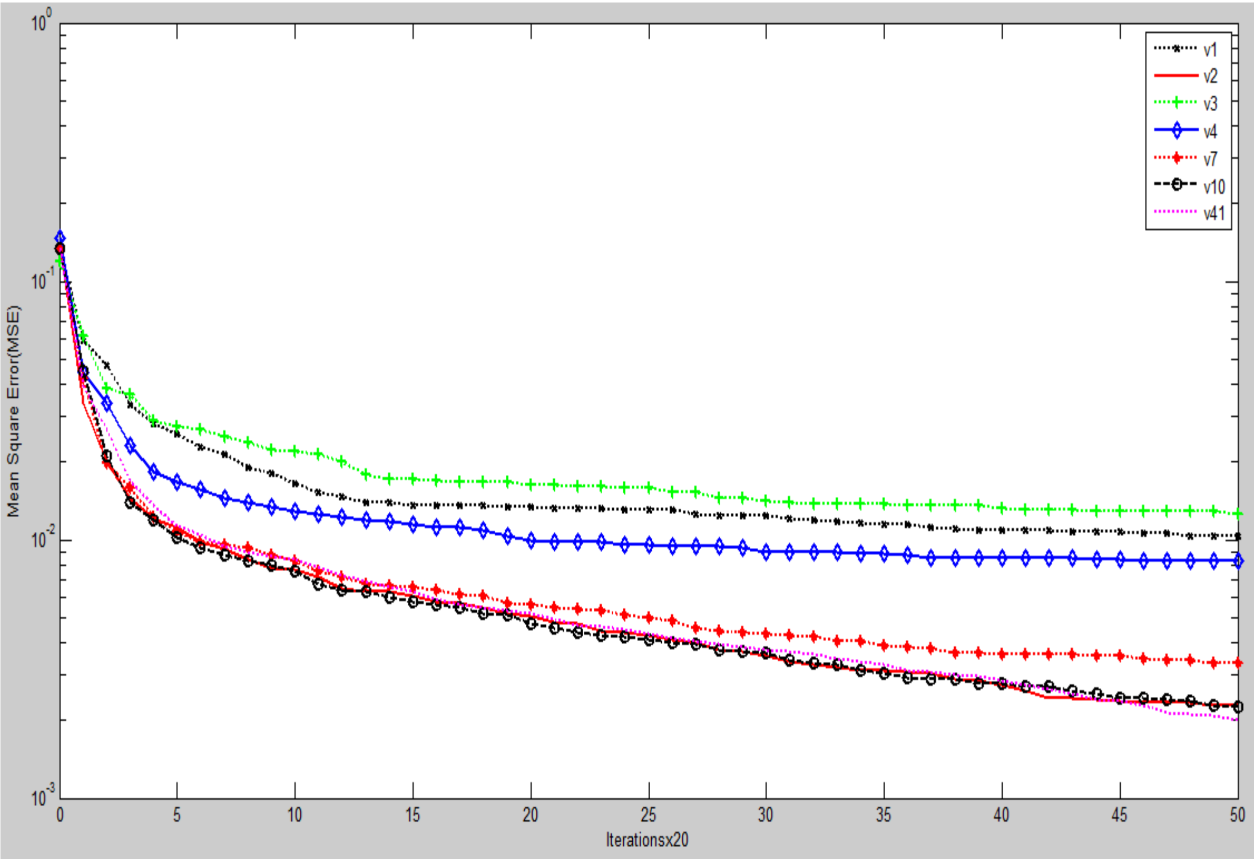


Fig.7.9 Banknote Training Error Convergence

MSE performance of proposed *"TSDE/bin" (V41)* & other mutation strategies for Banknote classification problem is given in Figure 7.9. This figure illustrates that overall performance of *"DE/best/1/bin"(V_2)*, *"DE/current to best/1/bin"(V_7)*, *"DE/rand repeated to best/1/bin"(V_{10})* and *"TSDE/bin"(V_{41})* is better than *"DE/rand/1/bin"(V_1)*, *"DE/rand/2/bin"(V_3)* & *"DE/best/2/bin"(V_4)*; however, the performance of *"TSDE/bin" (V41)* is best among all other mutation strategies from 4700 iteration till final iteration. It is important to note that performance of *"DE/rand/2/bin"(V_3)* has worst performance than all other mutation strategies for this problem.

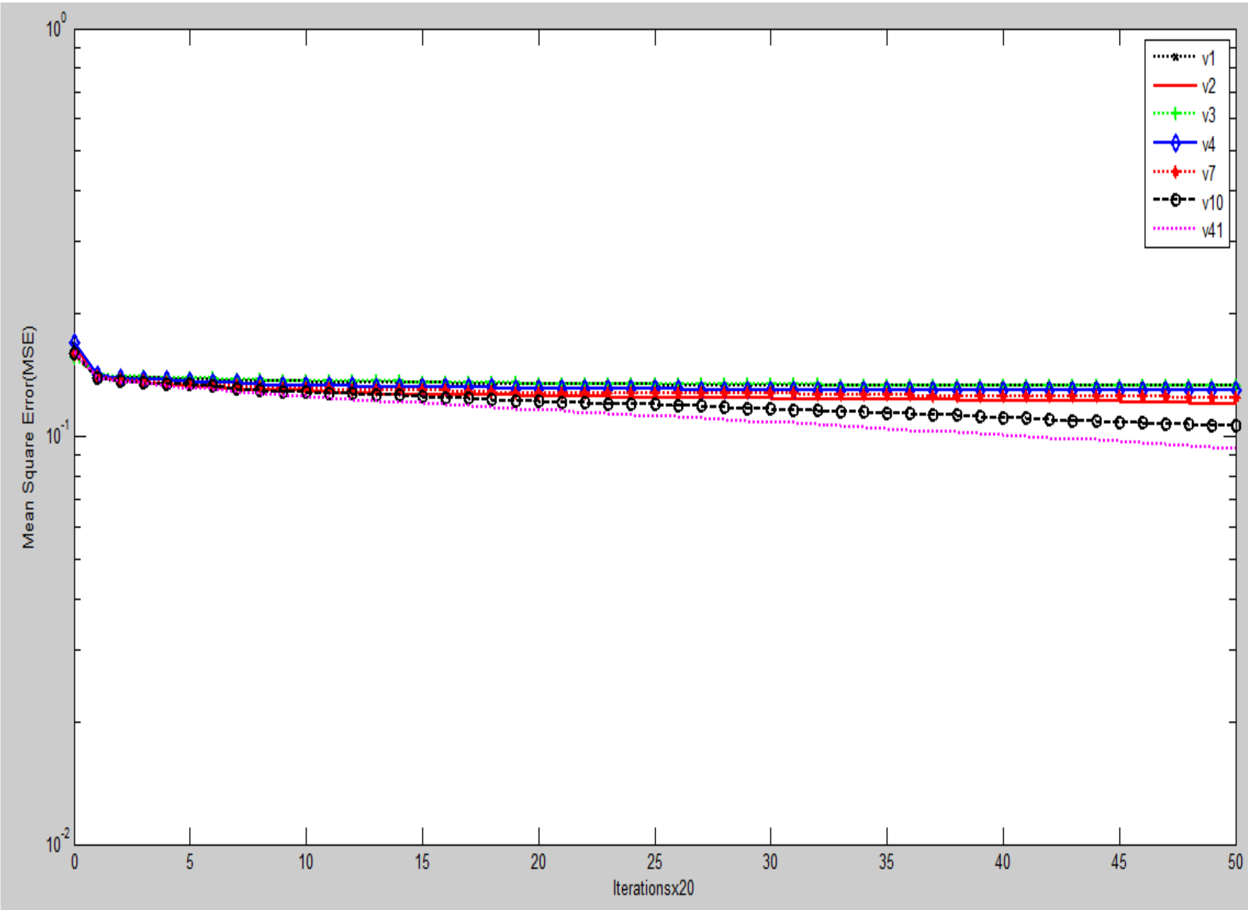


Fig.7.10 Diabetes Training Error Convergence

MSE performance of *"TSDE/bin" (V41)* and other mutilation strategies is graphically contained in Figure 7.10 for diabetes classification problem. This figure depicts that MSE of *"TSDE/bin" (V41)* is similar to other mutation strategies in earlier iterations; however, its performance starts boosting from 510th iteration to final iteration. It is obvious that performance of *"TSDE/bin" (V41)* is better among all mutation strategies in the specified iterations. It is important to note that performance of *"DE/rand repeated to best/1/bin"(V_{10})* has second best performance.

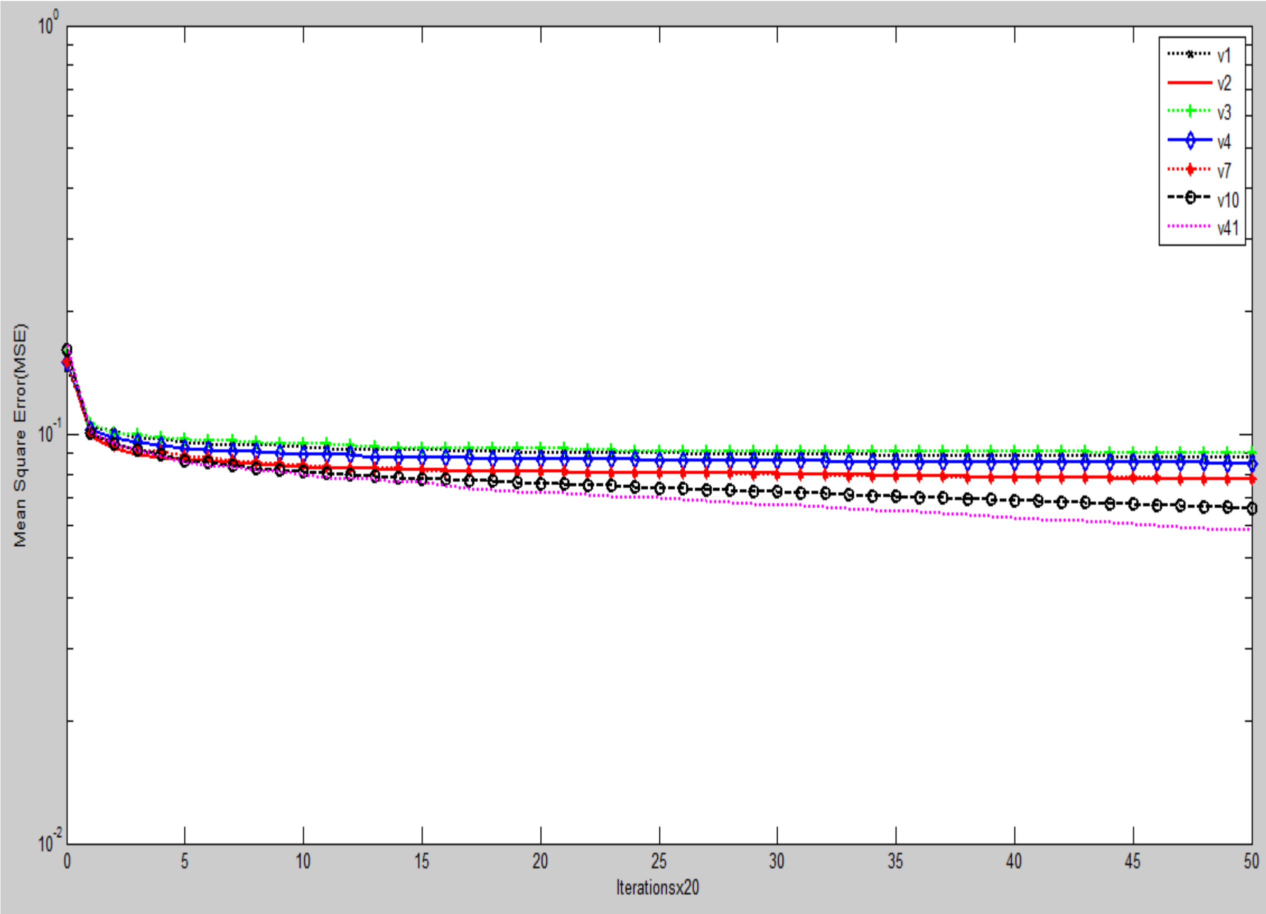


Fig.7.11 Mammographic Training Error Convergence

Figure 7.11 contains MSE performance of *“TSDE/bin” (V41)* & other mutation strategies for mammographic data classification problem. This figure shows that performance of *"DE/rand/2/bin"(V_3)*, *"DE/rand/1/bin"(V_1)* & *"DE/best/2/bin"(V_4)* remains almost similar in all iterations. The performance of *“TSDE/bin” (V41)* & *"DE/rand repeated to best/1/bin"(V_{10})* remains similar till 190th iteration; however, *“TSDE/bin” (V41)* starts gradually decreasing from 190th iteration to final iteration. Overall, it can be concluded that MSE performance of *“TSDE/bin” (V41)* is best among all mutation strategies while *"DE/rand repeated to best/1/bin"(V_{10})* holds second best MSE performance.

Figure 7.12 shows MSE of proposed *“TSDE/bin” (V41)* & other DE mutation strategies for customer classification data. It is obvious from this figure that performance of *"DE/best/1/bin"(V_2)*, *"DE/current to best/1/bin"(V_7)*, *"DE/rand repeated to best/1/bin"(V_{10})* and *“TSDE/bin” (V41)* is similar in early iterations however *“TSDE/bin” (V41)* has dominating performance among all mutation strategies from 260th iteration to final iteration. Mutation strategy *“TSDE/bin” (V41)* achieves MSE performance $10^{-1.8}$ in final iteration while *"DE/rand* *repeated to best/1/bin"(V_{10})* achieves value $10^{-1.6}$ in last iteration that secures second best performance.

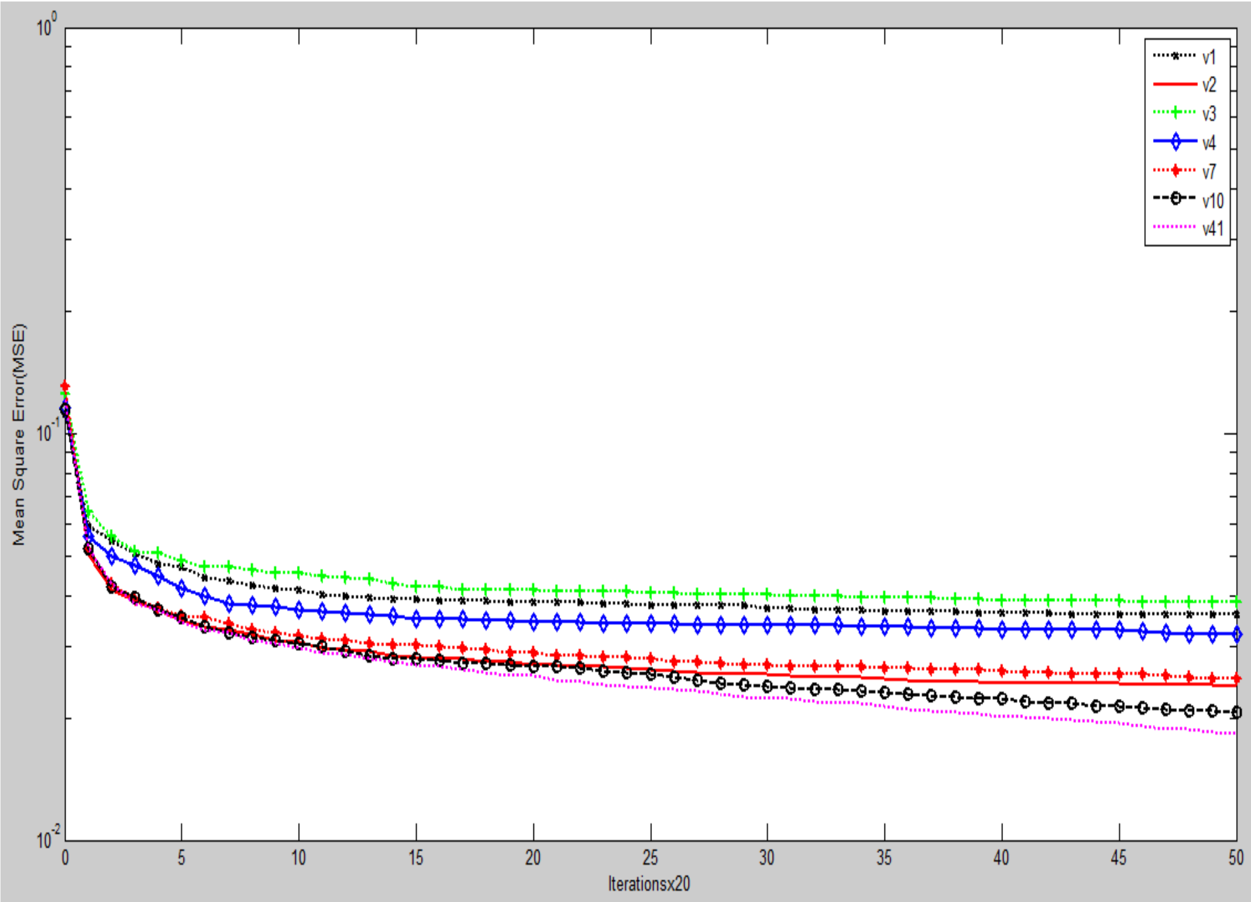


Fig.7.12 Customer Training Error Convergence

The training error convergence graphs of all datasets are contained in figures 7.4 to 7.13. Logarithmic convergence graphs of training error shows number of training iterations horizontally and mean square error (MSE) performance vertically. Training error convergence graphs contains five commonly used DE mutation strategies (V_1, V_2, V_3, V_4, V_7), the proposed mutation strategy *"TSDE/bin" (V41)* and one other better performing variants *"DE/rand repeated to best/1/bin"(V_{10})*. In essence, it is noted that the mean square error of *"TSDE/bin" (V41)* remains smallest in all experiments presented in above graphical form. Also, it is found that MSE value of *"TSDE/bin" (V41)* start at same momentum like other techniques; however it converges more quickly than other conventional techniques *"DE/rand/1/bin"(V_1) ,"DE/best/1/bin"(V_2), "DE/rand/2/bin"(V_3), "DE/best/2/bin"(V_4), "DE/current to best/1/bin"(V_7)*

and *"DE/rand repeated to best/1/bin"(V_{10})* in subsequent iterations in most of the cases. It is also noted that the mutation strategy *"DE/rand repeated to best/1/bin"(V_{10})* proves itself to be one of the better competing DE strategy; it is following *"TSDE/bin" (V41)* in most cases of convergence graphs. It is important to mention here that *"DE/rand repeated to best/1/bin"(V_{10})* strategy has never been declared to be one of the best performing variant or among top performing variants of DE algorithm. The convergence graph clearly shows the dominating performance of the proposed DE mutation variant *"TSDE/bin" (V41)*. Most of the researchers have concentrated on very few basic DE strategies like *"DE/rand/1/bin (V_1)"*, *"DE/best/1/bin (V_2), "DE/rand/2/bin (V_3)", "DE/best/2/bin (V_4)"* etc and some key performing mutation strategies remained unnoticed in the DE research. Thus it is concluded that *"TSDE/bin" (V41)* is more robust mutation strategies than other conventional mutation strategies.

7.10 Pseudocode of RCPDENN Training

Figure 7.14 shows the pseudocode of ANN training using Random controlled pool base Differential Evolution algorithm. The pseudocode contains implementation flow of ANN training using RCPDE algorithm with parameter pool and strategy candidate pool initialization. After initialization the fitness value/Mean Square Error (MSE) of each ANN architecture is calculated then ANN architectures are evolved by apply DE algorithm operator's mutation, crossover and selection. After evolutionary process the optimal ANN architecture is obtained. RCPDENN is also implemented through computer simulation.

The strategy candidate pool: “rand/1/bin”, “rand/2/bin”, “current-to-rand/1”, “rand to best /1/bin”, “TSDE/bin”

The parameter pool: [F=1.0, Cr=0.1], [F=0.8, Cr=0.2], [F=0.5, Cr=0.9], [F=0.7, Cr=0.5]

Set BPNN parameters:

- *Maximum training iterations*
- *Learning rate*
- *Hidden neurons*
- *Neurons in input later*
- *Neurons in output layer*

1. Generate the initial ANN architectures $P_G = \{X_{1,G}, \ldots\ldots X_{NP,G}\}$ for generation G=0 and randomly initialize weights for each ANN architectures $X_{i,G} = \{x^1_{i,G} \ldots\ldots\ldots x^D_{i,G}\}$ where *i* = 1…………NP

2. Randomly initialize control parameter memory from control parameter pool and strategy memory from strategy pool for each ANN architecture $X_{i,G}$ where *i* = 1…………NP

3. FOR *i* = 1 to NP

   Calculate mse $f(X_{i,G})$ for each ANN architectures $X_{i,G}$ using parameter pool value and mutation strategy assigned in step-2.

   END FOR

4. WHILE the stopping criterion is not true

Step 4.1 Architecture and Strategy selection

FOR *i* = 1 to NP

Step 4.1.1 Architectures selection

Select random ANN architectures to be used in the equation of mutation strategy ***S*** (given in equaions1-4) from current Population

Step 4.1.2 Mutation Step

For the i[th] target architecture $X_{i,G}$ generate a donor architecture $V_{i,G} = \{v^1_{i,G} \ldots\ldots v^D_{i,G}\}$ with i[th] strategy $\boldsymbol{S_i}$ from memory strategy and i[th] control parameter $\boldsymbol{F_i}$ from control parameter memory.

END FOR

Step 4.2 Crossover Step

FOR *i* = 1 to NP

For the i[th] target architecture $X_{i,G}$ generate trial architecture $U_{i,G} = \{u^1_{i,G} \ldots\ldots u^D_{i,G}\}$ with the specified crossover scheme using control parameter $\boldsymbol{CR_i}$ from control parameter memory.

END FOR

Step 4.3 Selection Step

FOR *i*=1 to NP

Evaluate the trial architecture $U_{i,G}$ against the target architecture $X_{i,G}$ with mse function *f*

IF $f(U_{i,G} \leq f\left(X_{i,G}\right)$, THEN $X_{i,G+1} = U_{i,G}$, $f(X_{i,G}) = f\left(U_{i,G}\right)$

IF $f(U_{i,G} \leq f\left(X_{best,G}\right)$, THEN $X_{best,G+1} = U_{i,G}$, $f(X_{best,G}) = f\left(U_{i,G}\right)$

END IF

ELSE

Strategy memory Updating

Randomly select a *mutation strategy* ***S*** from *strategy pool* and update control parameter memory for ith architecture

Control Parameter memory Updating

Randomly select a pair of *control parameters* (***F***, ***CR***) from *parameter pool* and update control parameter memory for ith architecture

END IF

END FOR

Step 4.4 Update mse

Step 4.5 *increment generation number G=G+1*

Step 5. END WHILE

Step 6 Get Optimum ANN architecture

Figure 7.13 *Pseudocode of ANN training algorithm using Random Controlled Pool base DE algorithm (RCPDENN)*

7.11 Flowchart of RCPDE-NN Training

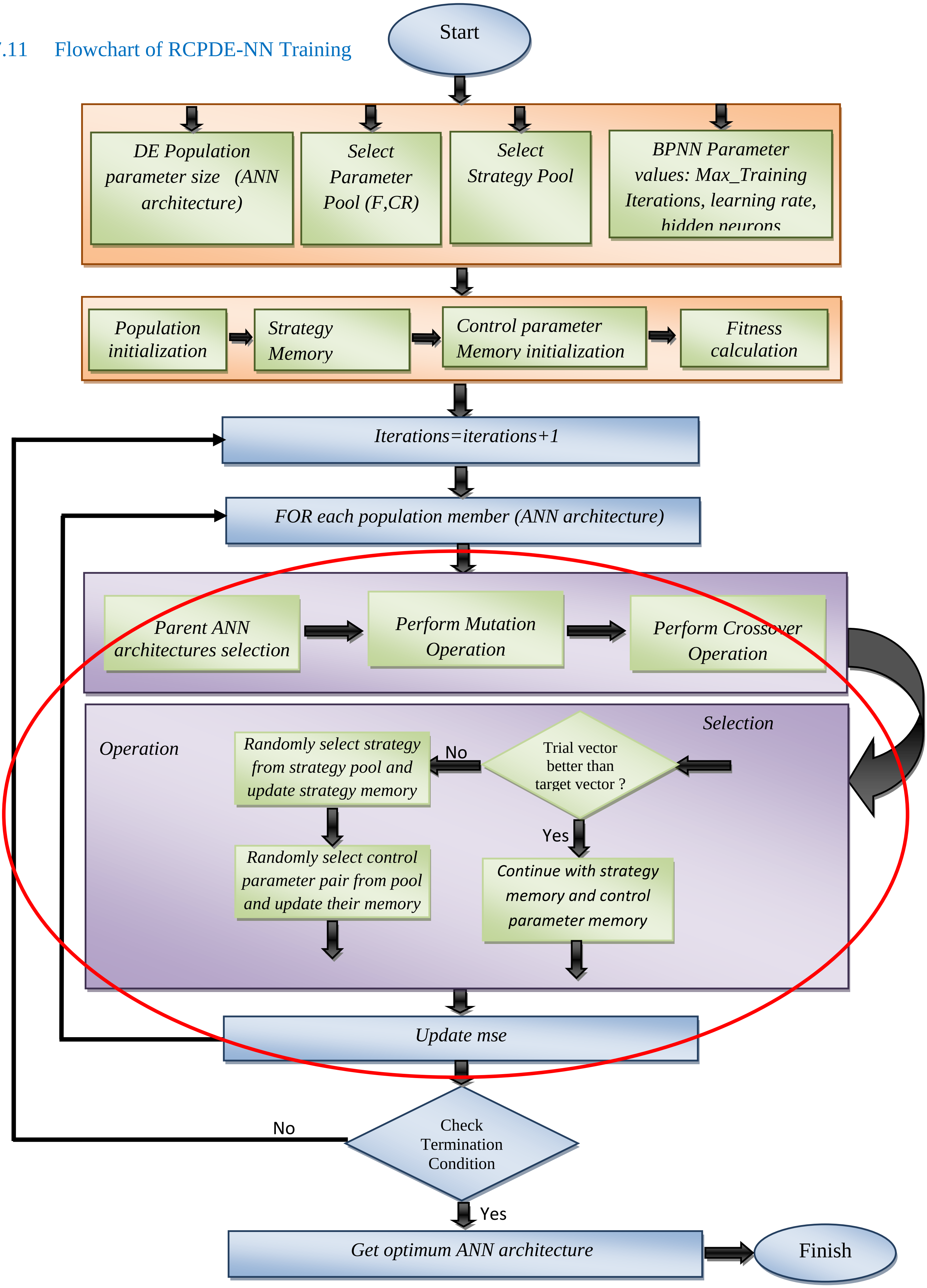


Figure 7.14 *Pseudocode of ANN training algorithm using Random Controlled Pool base DE algorithm (RCPDENN)*

7.12 RCPDENN accuracy of results

Table.7.4 Training Error (Mean ± Standard Deviation)

Data Set	DENN	CoDENN	EPSDENN	jDENN	RCPDENN
Iris	92.67±7.98	94.00±7.34	**96.67±4.71**	94.00±7.34	94.00±7.34
Balance Scale	70.71±14.48	70.87±16.06	71.43±14.98	72.14±15.84	**73.10±15.21**
Breast Cancer	96.57±2.10	**96.93±1.91**	96.21±2.43	95.64±2.22	96.50±2.11
Haberman	72.33±3.78	**73.00±1.05**	**73.33±0.00**	**73.33±0.00**	**73.33±0.00**
Pima Indian	75.52±3.14	72.40±4.65	73.25±4.02	71.56±3.59	**75.84±3.53**
Seeds	87.86±6.78	85.48±8.43	88.09±6.05	87.86±5.66	**88.33±7.05**
Vertebral 2-Column	75.97±13.59	**78.55±10.12**	77.58±12.41	78.39±12.54	77.90±11.30
Heart	80.56±8.01	77.78±6.23	80.19±5.92	80.00±7.40	**81.30±4.49**
Fertility	84.00±6.58	**87.22±4.41**	83.33±8.29	77.78±6.67	83.89±6.51
Bank Note Authentication	96.53±1.09	97.34±0.62	97.45±1.07	98.50±0.38	**98.54±0.73**
Diabetes	67.50±4.81	67.76±6.73	**70.52±5.06**	68.19±3.73	68.79±4.67
Mammographic	**81.61±2.38**	80.52±3.88	79.79±2.87	78.75±4.57	80.63±2.63
Customer	**91.36±4.39**	89.20±4.19	90.19±4.75	90.00±4.63	90.80±2.76
Wine	86.94±7.41	89.17±6.06	90.83±6.01	92.22±4.30	**94.44±3.21**

The accuracy of classification result shows that the proposed RCPDE has overall best performance among all other competitors. In most of the cases the performance of RCPDE is either better or comparable to algorithms reported in table 7.4. The results clearly indicated that RCPDE successful trains ANN and has significant performance.

7.13 RCPDE Training Error (Mean ± Standard Deviation)

Table.7.5 Training Error (Mean ± Standard Deviation)

	DENN	CoDENN	EPSDENN	jDENN	RCPDENN
Iris	1.59E-02±4.73E-03	7.20E-03±1.84E-03	8.17E-03±2.18E-03	3.45E-03±1.72E-03	9.15E-03±3.96E-03
Balance Scale	3.05E-02±4.73E-03	2.85E-02±2.01E-03	2.62E-02±2.40E-03	2.19E-02±2.50E-03	1.93E-02±3.82E-03
Breast Cancer	1.09E-02±1.83E-03	9.77E-03±2.06E-03	8.95E-03±2.04E-03	6.49E-03±1.87E-03	6.62E-03±2.70E-03
Haberman	6.62E-02±2.32E-02	5.97E-02±2.14E-02	5.73E-02±2.11E-02	4.73E-02±2.03E-02	5.77E-02±2.17E-02
Pima Indian	1.12E-01±2.92E-03	1.09E-01±2.11E-03	1.04E-01±1.83E-03	8.94E-02±3.15E-03	7.88E-02±1.40E-02
Seeds	2.71E-02±3.67E-03	1.09E-01±2.11E-03	1.16E-04±3.66E-04	8.94E-02±3.15E-03	7.88E-02±1.40E-02
Vertebral 2-Column	6.15E-02±7.55E-03	1.09E-01±2.11E-03	1.16E-04±3.66E-04	8.94E-02±3.15E-03	7.88E-02±1.40E-02
Heart	5.86E-02±5.83E-03	5.80E-02±5.36E-03	5.07E-02±4.54E-03	4.11E-02±5.68E-03	2.67E-02±4.22E-03
Fertility	3.38E-02±5.47E-03	2.69E-02±5.07E-03	2.13E-02±6.53E-03	1.30E-02±6.18E-03	1.78E-02±7.16E-03
Bank Note Authentication	1.04E-02±1.76E-03	7.72E-03±1.17E-03	5.33E-03±1.30E-03	1.27E-03±1.42E-03	1.62E-03±1.49E-03
Diabetes	1.33E-01±2.63E-03	1.28E-01±1.86E-03	1.26E-01±2.29E-03	1.10E-01±5.08E-03	1.23E-01±6.00E-03
Mammographic	2.32E-03±3.24E-03	8.39E-02±2.43E-03	8.11E-02±3.37E-03	7.33E-02±3.11E-03	7.67E-02±6.02E-03
Customer	3.61E-02±4.72E-03	3.17E-02±4.62E-03	2.91E-02±4.45E-03	2.22E-02±4.23E-03	1.74E-02±3.52E-03
Wine	1.80E-02±4.49E-03	1.23E-02±3.21E-03	2.87E-03±1.57E-03	1.16E-04±3.66E-04	1.16E-04±3.66E-04

The Mean Square Error results of RCPDENN, DENN, jDENN, CoDENN and EPSDENN are shown in table 7.5. The convergence graphs of MSE are shown as follows.

7.14 RCPDE Error Convergence Graphs

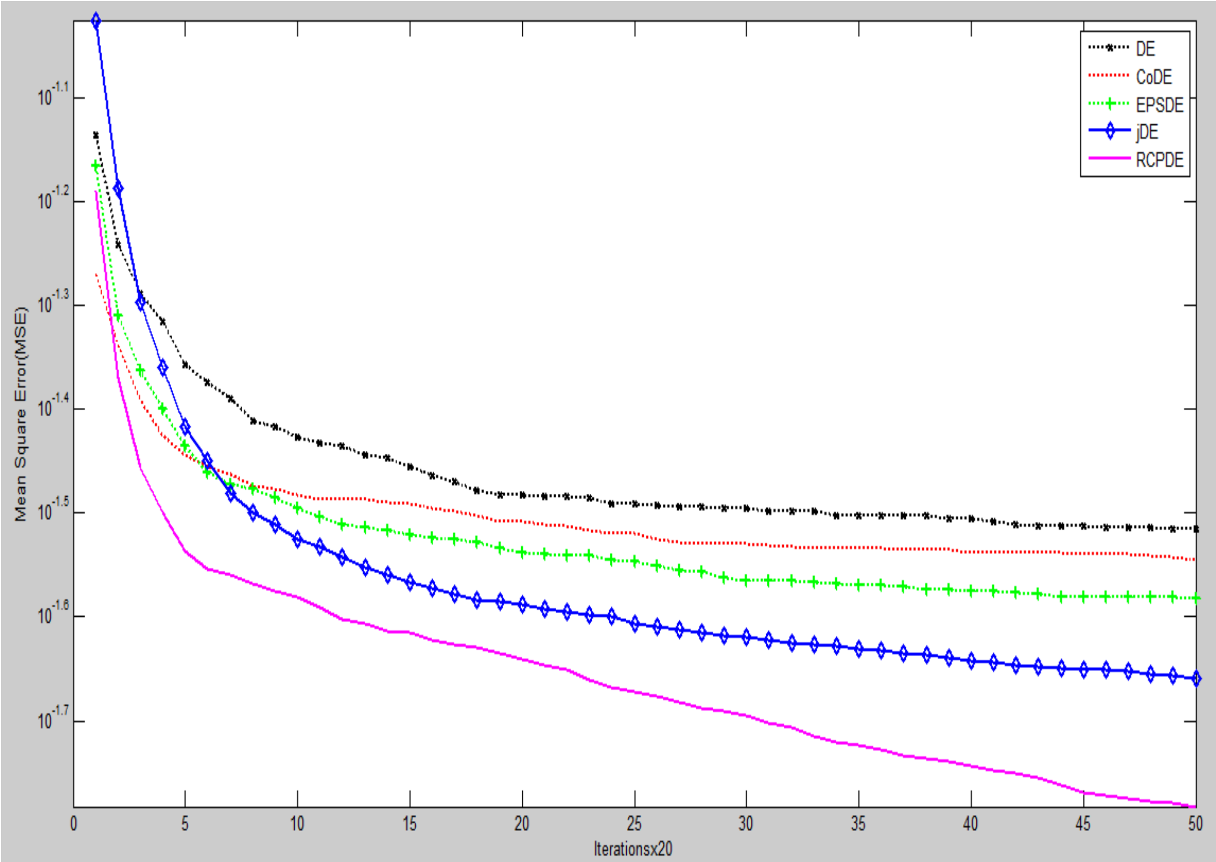


Figure 7.15 Balance Scale Training Error Convergence

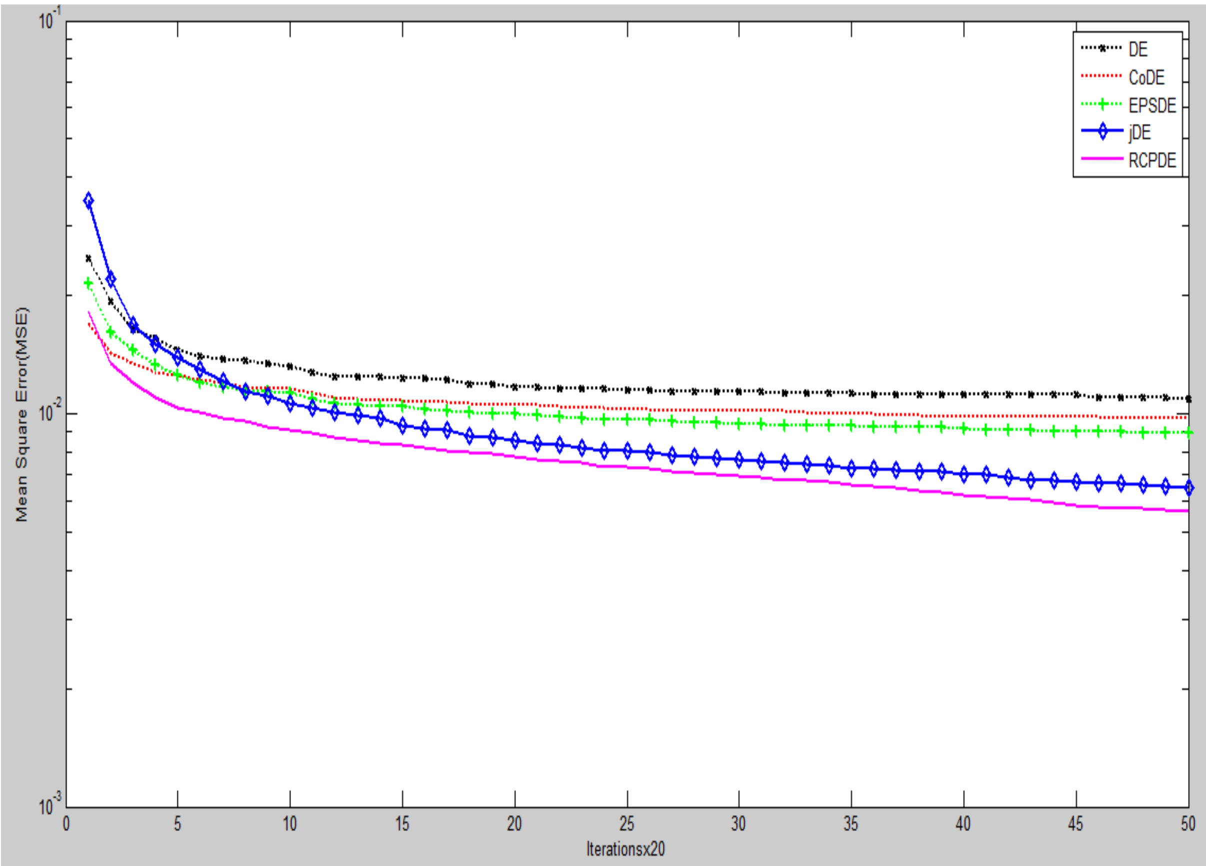


Figure 7.16 Breast Cancer Training Error Convergence

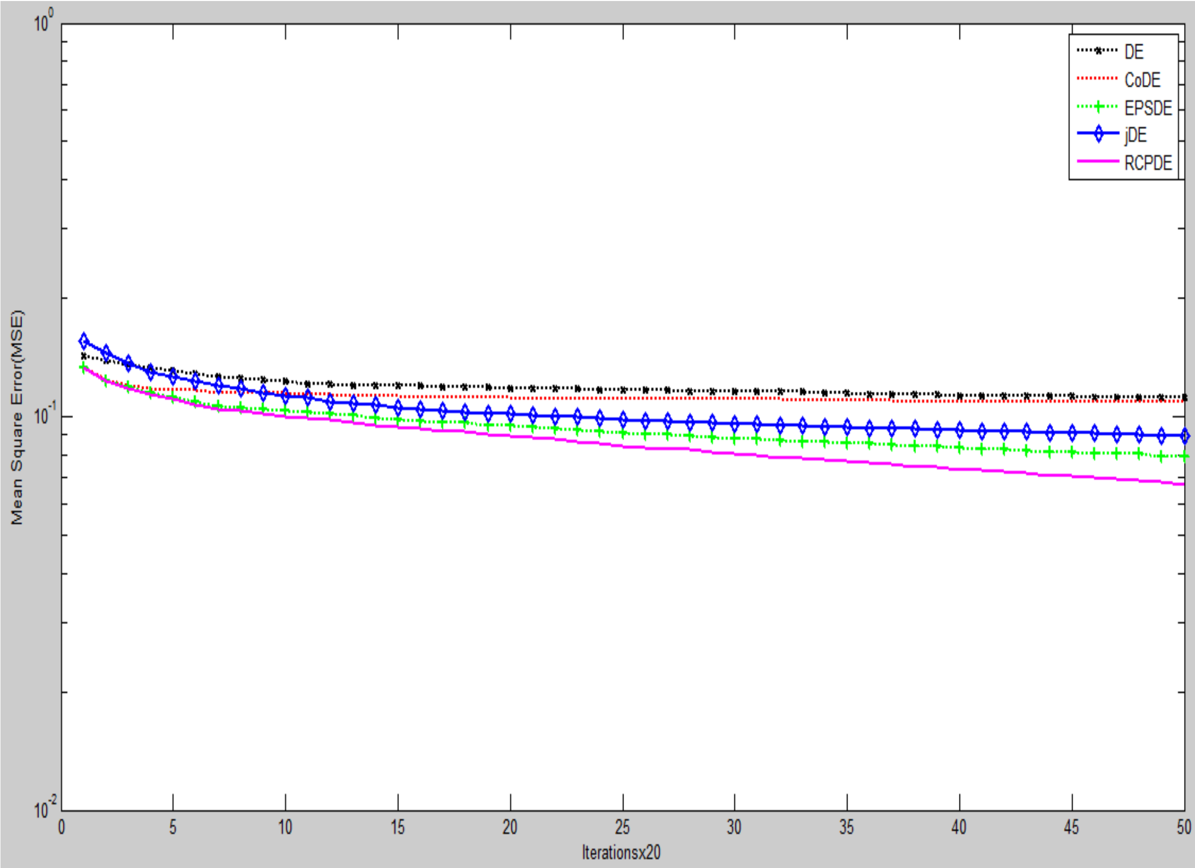


Figure 7.17 Pima Indian Training Error Convergence

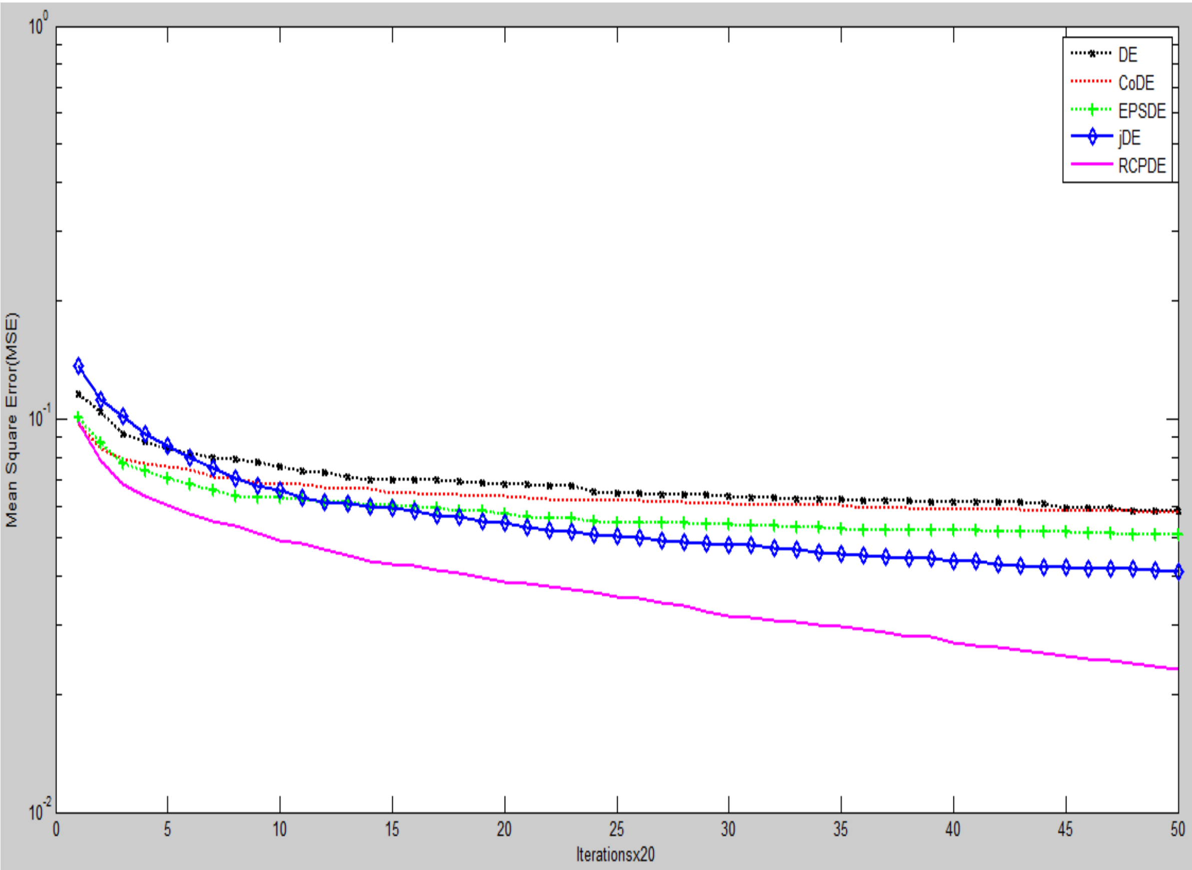


Figure 7.18 Heart Training Error Convergence

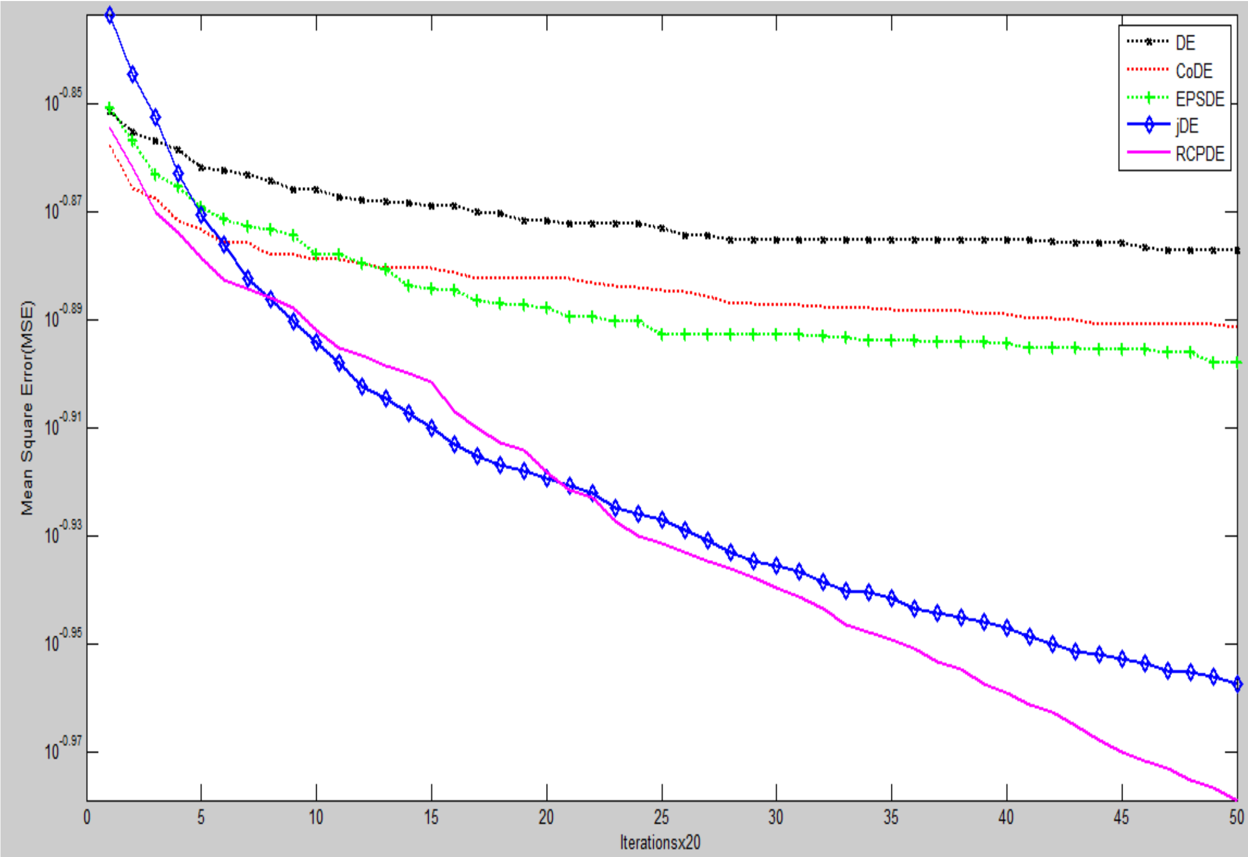


Figure 7.19 Diabetes Training Error Convergence

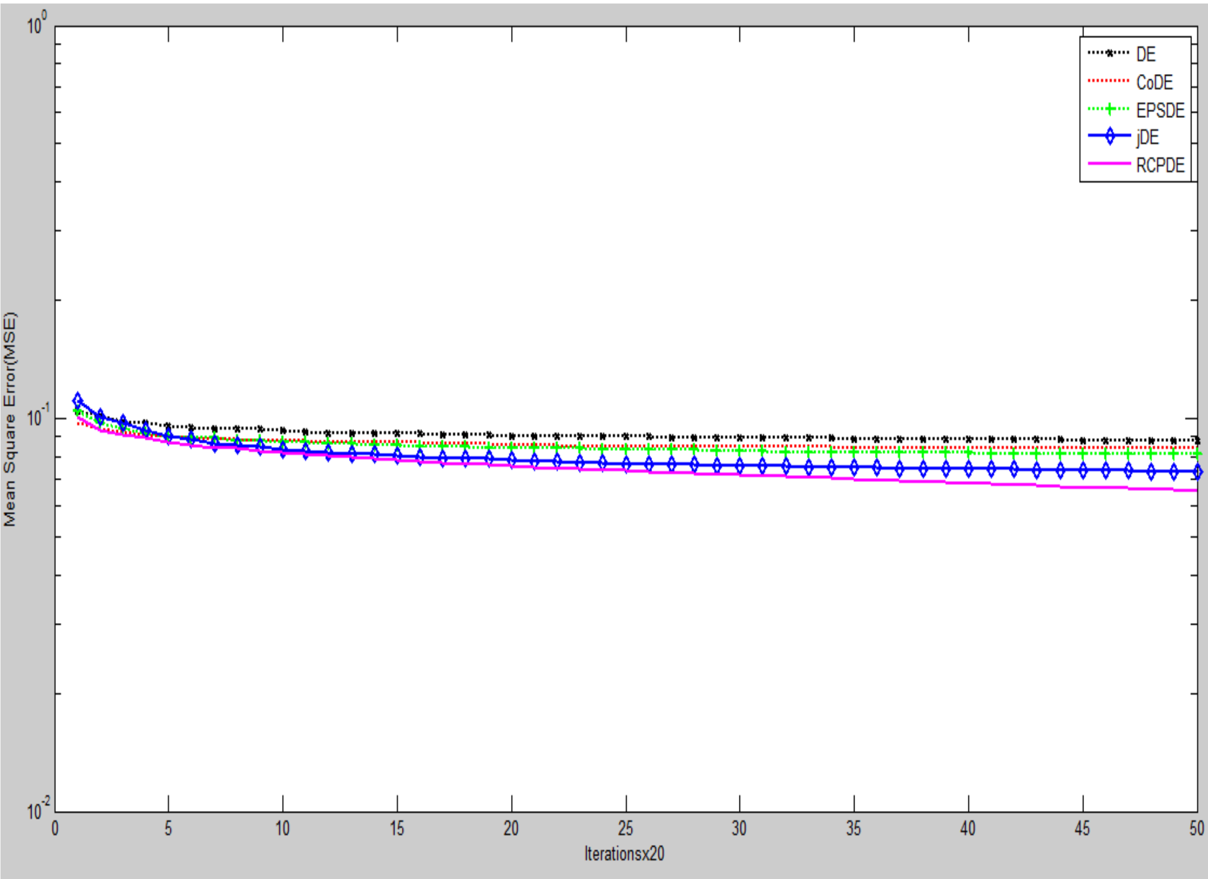


Figure 7.20 Mammographic Training Error Convergence

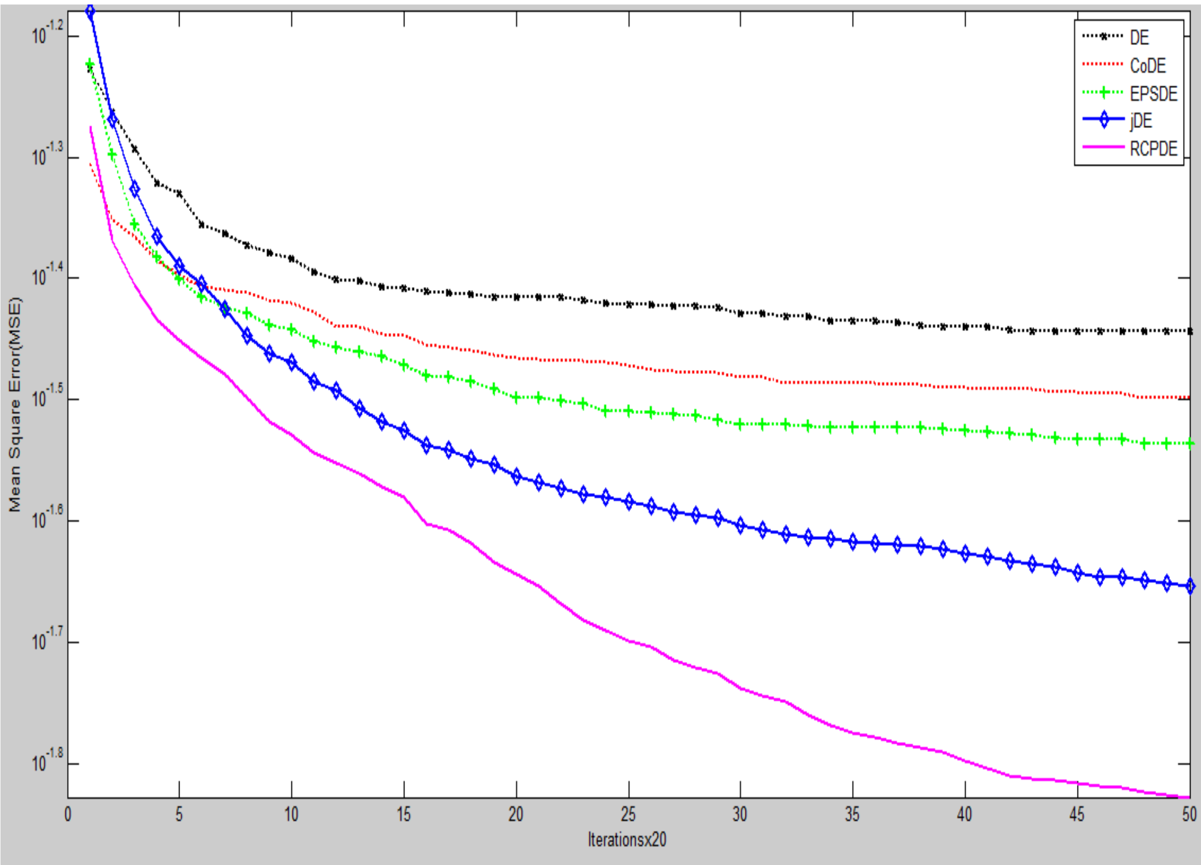


Figure 7.21 Customer Training Error Convergence

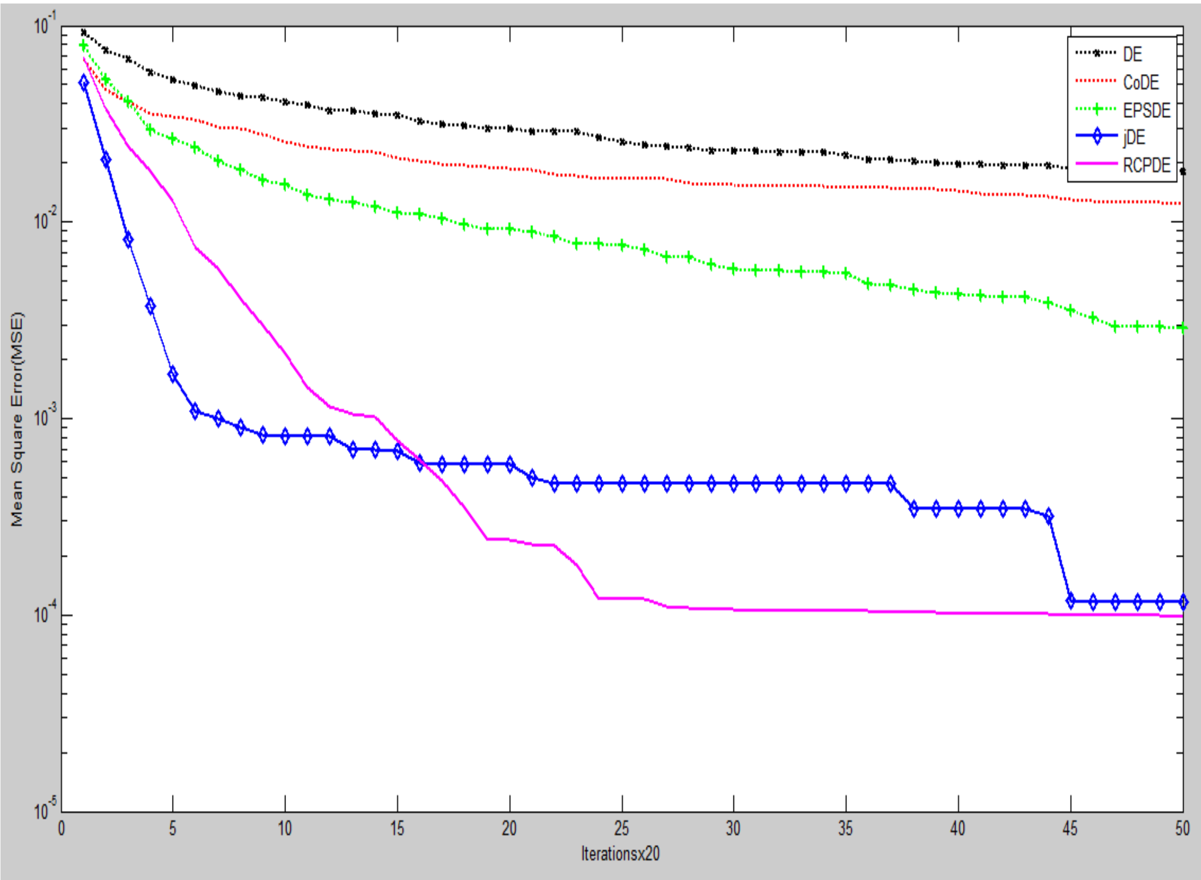


Figure 7.22 Wine Training Error Convergence

The training error convergence graphs of some datasets are contained in Figures 7.16 to 7.23. Logarithmic convergence graphs of training error shows number of training iterations horizontally and mean square error (MSE) performance vertically. Training error convergence graphs contains proposed RCPDE algorithm, standard DE algorithm and three other well known state of the art algorithms jDE, EPSDE and CoDE. In essence, it is noted that the mean square error of proposed RCPDE remains smallest in all experiments presented in above graphical form. Also, it is found that MSE value of RCPDE start at same momentum like other algorithms; however it converges more quickly than DE, jDE, EPSDE and CoDE in subsequent iterations in most of the cases. The convergence graph clearly shows the dominating performance of the proposed RCPDE algorithm. Thus it is concluded that RCPDE is more robust than other algorithm presented in this section. The random selection of control parameters and mutation strategies makes proposed algorithm simple and requires less computations.

Chapter # 8: Conclusion and Future Work

8.0 Chapter Summary

This chapter presents conclusion and future work of the research work. Advantages, limitations of the proposed system along with its application to ANN are also presented here.

8.1Conclusion

Two novel advancements in DE algorithm have been proposed in this thesis which are presented in chapter 5 and 6. The proposed advancements are helpful to improve the convergence speed and solution quality of DE algorithm. Moreover an effort is made to make a set of DE mutation strategies consistent. The main contributions in this thesis discussed as follows:

8.1.1 Tournament selection based advancement in DE algorithm

The main contribution to this work is to propose a new mutation strategy of DE algorithm. The proposed novel tournament base parent selection base mutation strategy was thoroughly analyzed and implemented for function optimization application. The TSDE mutation strategy has been compared with many other similar algorithms. During the comparison standard benchmark functions are used. Simulation results which were presented in chapter 5 have shown that the proposed model performed far better than others. The proposed *TSDE* mutation strategy has dominating performance among the DE mutation strategies. This new mutation strategy *TSDE* will prove to be a valuable addition to DE literature. A comprehensive set of well known N-dimensional benchmark functions have been used to evaluate the performance of the proposed *TSDE* as well as existing mutation strategies of DE algorithm. In this research, extensive comparison is performed by considering the most number of mutation strategies; such a comparison is missing in the literature. This comparison unveils some very beneficial facts related to the DE research improvements. The other important aspect of this research is the statistical comparison of the DE mutation strategies; such a comparison is also not available in the literature. Since total strategies are forty two and it is not possible to report all the mutation strategies in few tables, so proposed TSDE mutation strategy, five commonly used mutation strategies and another one of the best performing mutation strategy are reported in the results. It can be concluded that there is a deviation in the performance of DE algorithm variants and the selection of DE variant affects the performance result of DE algorithm. Better results can be obtained by choosing the best performing strategies. The proposed *"TSDE"* has the leading performance among the DE mutation strategies for both fitness value and number of function call performance parameters, mutation strategy *"DE/rand to best/1/bin(V_{10})", "V_7"*

and *“DE/best/1/bin (V_2)”* are the other better performance mutation strategies. This research work also reveals one of the best performing mutation strategy *“DE/rand to best/1/bin (V_{10})”* that has been rarely brings into play. This research work will prove to be a significant addition in DE literature. The results of TSDE are also compared with two well known heuristics GA and PSO that shows dominating performance of TSDE.

8.1.2 Random Controlled Pool based advancement in DE algorithm

This research work introduces random controlled pool based selection differential evolution algorithm (RCPDE). A new mutation strategies pool and control parameters pool are used in RCPDE that are helpful in balancing the exploration and exploitation ability of DE algorithm by incorporating potential mutation strategies and diverse control parameter values in DE algorithm. The two commonly used performance metrics NFC and Average fitness values are used to compare the performance of the proposed DE algorithm with other similar state of the art DE algorithms. Simulation results presented in chapter 6 have shown that RCPDE has better performance than other algorithms. A comprehensive set of N-dimensional benchmark functions is used to evaluate the performance of RCPDE. Research result shows that RCPDE improves the solution quality as well as convergence speed.

8.1.3 DE mutation strategies inconsistency identification and removal

Selection of DE variant affects the performance result of DE algorithm since there is a deviation in performance of DE variants. DE algorithm variants in the literature have naming and mathematical equation inconsistencies. This research work identifies these inconsistencies and presents a consistent set of DE variants. The detail of naming and formulation inconstancies is discussed in chapter 4 of this thesis. Naming and formulation inconsistencies in DE variants are removed in section-V. The naming and formulation inconsistencies are removed based on the number of vectors and order of vectors used to form the equation of the variant. This research work will prove to be a significant addition to the DE literature in view of the fact that the existence of any inconsistency might trigger off new researchers. The further studies are still required to explore the limitations, advantages and flaws in this direction. The main focus of this work is not the overwhelming the existing DE variants but new tracks to work on the DE variants in optimization

8.1.4 Validation of proposed Techniques

The proposed techniques are validated to train artificial neural network for data classification. Some commonly used and well know data classification problems are used to validate the performance of proposed TSDE mutation strategy & RCPDE algorithm. The research results of TSDE mutation strategy are compared with conventional mutation strategies. Research results of RCPDE are compared with standard DE algorithm and some other state of the art algorithms like jDE, EPSDE and CoDE. The research result based on testing accuracy, training error and mean square error convergence graphs shows that the performance of TSDE is better than all other mutation strategies & RCPDE algorithm has dominating performance than other state of the art algorithms.

8.2 Direction for Future Work

The main focus of this work is not the overwhelming the existing DE variants but new tracks to work on the DE variants in optimization. Future work of this research work is to develop more powerful parent selection schemes that may improve the performance of DE algorithm. The further studies are still required to explore the limitations, advantages and flaws in this direction.

Another possible direction of future work of this research work is to incorporate the memorization concept of convergence track of mutation strategies and control parameters over time.

References

[1] T. Sum-Im, "Self-adaptive differential evolution algorithm for economic dispatch with transmission losses consideration," in *In proceeding of IEEE conference TENCON*, Fukuoka, 2010, pp. 90-95.

[2] P. Rocca, G. Oliveri, and A. Massa, "Differential Evolution as Applied to Electromagnetics," *IEEE Antennas and Propagation Magazine*, vol. 53, no. 1, pp. 38-49, 2011.

[3] M. D. Kapadi and R. D. Gudi, "Optimal control of fed-batch fermentation involving multiple feeds using Differential Evolution," *Process Biochemistry*, vol. 39, no. 11, pp. 1709-1721, 1709-1721 2004.

[4] N. Noman and H. Iba, "Inferring gene regulatory networks using differential evolution with local search heuristics," *ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 4, pp. 634-647, 2007.

[5] B. V. Babu and R. Angira, "Modified differential evolution (MDE) for optimization of non-linear chemical processes," *Computers & Chemical Engineering*, vol. 30, no. 6-7, pp. 989–1002, 2006.

[6] G. Schaefer and L. Nolle, "Optimal image colour extraction by differential evolution," *International Journal of Bio-Inspired Computation*, vol. 2, no. 3-4, pp. 251-257, May 2010.

[7] G. D. Magoulas, V. P. Plagianakos, and M. N. Vrahatis, "Neural network-based colonoscopic diagnosis using on-line learning and differential evolution," *Applied Soft Computing*, vol. 4, no. 4, pp. 369-379, 2004.

[8] V. Tirronen and et al., "An Enhanced Memetic Differential Evolution in Filter Design for Defect Detection in Paper Production," *Evolutionary Computation Journal*, vol. 16, no. 4, pp. 529-555, 2008.

[9] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-Based Differential Evolution," *IEEE Transaction of Evolutionary Computing*, vol. 12, no. 1, pp. 64-79, 2008.

[10] H.R. Tizhoosh, "Opposition-Based Learning: A New Scheme for Machine Intelligence," in *International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, Vienna, 2005, pp. 695- 701.

[11] M. Ali, M. Pant, and V. P. Singh, "An Improved Differential Evolution Algorithm for Real Parameter Optimization Problems," *International Journal of Recent Trends in Engineering*, vol. 1, no. 5, pp. 63-65, May 2009.

[12] S. Das and et. al., "Differential Evolution Using a Neighborhood-Based Mutation Operator," vol. 13, no. 3, pp. 526-553, 2009.

[13] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization An overview," *Information Processing Letters*, vol. 1, no. 1, pp. 33-57, 2007.

[14] A. K. Palit and D. Popovic, *Computational Intelligence in Time Series Forecasting*, M. J. Grimble, Ed. USA: Springer-Verlag, 2005.

[15] M. Mitchel, *An Introduction to Genetic Algorithms*. London: MIT Press, 1999.

[16] T. Weise, *Global Optimization Algorithms Theory and Application*., 2009.

[17] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*. USA: Morgan Kaufmann division of Academic Press, 2001.

[18] A. P. Engelbrecht, *Computational intelligence an introduction*. England: John Wiley & Sons, 2007.

[19] A.P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. West Sussex, England: John Wiley and Sons, 2005.

[20] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, Second Edition ed. New Jersey: JOHN WILEY & SONS, 2004.

[21] G. C. Onwubolu and D. Davendra, *Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization*, J. Kacprzyk, Ed. Poland: Springer, 2009.

[22] 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.

[23] M M Ali, C Khompatraporn, and Z B Zabinsky, "A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems," *Journal of Global Optimization*, pp. 635–672, 2005.

[24] Brest J, Greiner S, Boskivic B, Mernik M, and Zumer V, "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.

[25] 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.

[26] 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, 2005.

[27] Miliie Pant, Radha Thangaraj, and V P Singh, "A New Differential Evolution Algorithm for Solving Global Optimization Problems," in *International Conference on Advanced Computer Control*, Singapore, 2009, pp. 388-392.

[28] J Adeyemo, F Bux, and F Otieno, "Differential evolution algorithm for crop planning: Single and multi-objective optimization model," *International Journal of the Physical Sciences*, vol. 5, no. 10, pp. 1592-1599, September 2010.

[29] X Yao, Y Liu, and G Lin, "Evolutionary Programming Made Faster," *IEEE Transaction on Evolutionary Computation*, vol. 3, no. 2, pp. 82-102, July 1999.

[30] J. Aalto and J. Lampinen, "A Population Adaptation Mechanism for Differential Evolution Algorithm," in *IEEE Symposium Series on Computational Intelligence*, 2015, pp. 1514-1521.

[31] S. L. Wang and et al., "Self-adapting approach in parameter tuning for differential evolution," in *Conference on Technologies and Applications of Artificial Intelligence*, 2015, pp. 113-119.

[32] J. Zhang and A. C. Sanderson, *Adaptive differential evolution: a robust approach to multimodal problem optimization*.: Springer, 2009, vol. Vol.1.

[33] 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.

[34] 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.

[35] 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.

[36] S. Aixin and et al., *Advances of computational intelligence in industrial systems*. Springer: Springer, 2008, vol. 116.

[37] V. Feoktistov, *Differential evolution: In Search of Solutions*. USA: Springer, 2006.

[38] J. Adeyemo, F. Bux, and F. Otieno, "Differential evolution algorithm for crop planning: Single and multi-objective optimization model," *International Journal of the Physical Sciences*, vol. 5,

no. 10, pp. 1592-1599, September 2010.

[39] 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.

[40] M. Ali, M. Pant, and A. Abraham, "Simplex Differential Evolution," *Acta Polytechnica Hungarica*, vol. 6, no. 5, pp. 95-115, 2009.

[41] L. Buhry and et. al., "New variants of the Differential Evolution Algorithm: Application for Neuroscientists," in *17th European Signal Processing Conference*, Scotland, 2009, pp. 2352-56.

[42] 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.

[43] E. K. P. Chong and S. H. Zak, *An introduction to optimization*, 2nd ed. New York, USA: John Wiley & Sons, 2008.

[44] C. Ting and C. Huang, "Varying Number of Difference Vectors in Differential Evolution," in *IEEE Congress on Evolutionary Computation 2009*, Chiayi, 2009, pp. 1351 - 1358.

[45] S. Hanuman and et. al., "Data Clustering Using Almost Parameter Free Differential Evolution Technique," *International Journal of Computer Applications*, vol. 8, no. 13, pp. 1-7, October 2010.

[46] R. Storn, "On the usage of differential evolution for function optimization," in *In Proceedings of IEEE NAFIPS'96*, Berkeley, 1996, pp. 519-523.

[47] G. Sudhakar and et. al., "Effective Image Clustering with Differential Evolution Technique," *International Journal of Computer and Communication Technology*, vol. 2, no. 1, pp. 11-19, 2010.

[48] J. Brest and et. al., "Performance comparison of self-adaptive and adaptive differential evolution algorithms," *Soft Computing*, vol. 11, pp. 618-629, 2007.

[49] K. Zielinski and et. al., "Choosing Suitable Variants of Differential Evolution and Particle Swarm Optimization for the Optimization of a PI Cascade Control," in *11th International Conference on Optimization of Electrical and Electronic Equipment*, , Bremen , 2008, pp. 55 - 62.

[50] B. Xin and et. al., "Hybridizing Differential Evolution and Particle Swarm Optimization to Design Powerful Optimizers: A Review and Taxonomy," *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 1 - 24, 2011.

[51] C. Veenhuis and M. Koppen, "Differential Evolution with Polymorphic Schemes," in *IEEE International Conference on Systems, Man, and Cybernetics* , San Antonio, 2009, pp. 1814-1819.

[52] S. Das and et. al., "Automatic Circle Detection on Images with Annealed Differential Evolution," in *8th International Conference on Hybrid Intelligent Systems* , 2008, pp. 684-689.

[53] Y. Biao, Z. Zengke, and S. Zhengshun, "Computing Nonlinear LTS Estimator Based on a Random Differential Evolution Strategy," *Tsinghua Science and Technology*, vol. 13, no. 1, pp. 59-64, February 2008.

[54] K.V. Price, "Differential Evolution: A Fast and Simple Numerical Optimizer ," in *Conference of the North American Fuzzy Information Processing Society*, Biennial, 1996, pp. 524 - 527.

[55] W. Gong and Z. Cai, "Adaptive Parameter Selection for Strategy Adaptation in Differential Evolution for Continuous Optimization," *Journal of Computers*, vol. 7, no. 3, pp. 672-679, March 2012.

[56] M. M. Farsangi and H. Nezamabadi-pour, "Differential evolutionary algorithm for allocation of

svc in a power system," *International Journal on Technical and Physical Problems of Engineering: Transaction on Power Engineering*, vol. 1, no. 1, pp. 15-22, December 2009.

[57] W. Gong, A. Fialho, and Z Cai, "Adaptive Strategy Selection in Differential Evolution," in *Genetic and Evolutionary Computation Conference(GECCO)*, Portland,USA, 2010.

[58] G. Jeyakumar and C. S. Velayutham, "An Empirical Performance Analysis of Differential Evolution Variants on Unconstrained Global Optimization Problems," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 2, pp. 077-086, 2010.

[59] V. L. Huang, A. K. Qin, and P. N. Suganthan, "Self-adaptive Differential Evolution Algorithm for Constrained Real-Parameter Optimization," in *EEE Congress on Evolutionary Computation*, Vancouver, 2006, pp. 17-24.

[60] A. Mandal and et. al., "Efficient circular array synthesis with a memetic differential evolution algorithm," *Progress In Electromagnetics Research*, vol. 38, p. 367

[61] J. Zhang and A. C. Sanderson, "JADE: Self-adaptive differential evolution with fast and reliable convergence performance," in *IEEE Congress on Evolutionary Computation 2007*, Singapore, 2007, pp. 2251 - 2258.

[62] A. Mandal and et. al., "Modified Differential Evolution with Local Search Algorithm for Real World Optimization," in *IEEE Congress on Evolutionary Computation 2011*, New Orleans, 2011, pp. 1565 - 1572.

[63] M. G. Epitropakis, V. P. Plagianakos, and M. N. Vrahatis, "Evolving cognitive and social experience in Particle Swarm Optimization through Differential Evolution," in *IEEE Congress on Evolutionary Computation*, Rion, 2010, pp. 1 - 8.

[64] A. Chowdhury and et. al., "Optimization of antenna configuration with a fitness-adaptive differential evolution algorithm," *Progress In Electromagnetics Research*, vol. 26, pp. 291-319, October 2010.

[65] S. Roy and et. al., "An adaptive differential evolution algorithm for autonomous deployment and localization of sensor nodes," *Progress In Electromagnetics Research*, vol. 29, pp. 289-309, April 2011.

[66] 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 , October 2009.

[67] U. Halder and et. al., "Self Adaptive Cluster Based and Weed Inspired Differential Evolution Algorithm For Real World Optimization," in *IEEE Congress on Evolutionary Computation*, Kolkata, 2011 , pp. 750 - 756.

[68] S. Dasgupta and et. al., "On Stability and Convergence of the Population-Dynamics in Differential Evolution," *AI Communications*, vol. 22, no. 1, pp. 1-20, January 2009.

[69] S. Ghosh and et. al., "Peak-to-Average Power Ratio Reduction in OFDM Systems Using an Adaptive Differential Evolution Algorithm," in *IEEE Congress on Evolutionary Computation*, Kolkata, 2011 , pp. 1941 - 1949.

[70] A. K. Qin and P. N. Suganthan, "Self-adaptive Differential Evolution Algorithm for Numerical Optimization," in *IEEE Congress on Evolutionary Computing*, Singapore , 2005, pp. 1785-1791.

[71] C. G. Martinez, F. J. Rodriguez, and M. Lozano, "Role differentiation and malleable mating for differential evolution: an analysis on large-scale optimisation," *Soft Computing*, vol. 15, pp. 2109-2127, 2011.

[72] W. Gong, Z. Cai, and H. Li, "Generalized hybrid generation scheme of differential evolution for global numerical optimization," *International Journal of Computational Intelligence and Applications*, vol. 10, no. 1, pp. 35-65 , 2011.

[73] W. Nakawiro and I. Erlich, "A new adaptive differential evolution algorithm for voltage stability constrained optimal power flow," in *17th Power Systems Computation Conference* , Stockholm, 2011, pp. 1-7.

[74] F. Peng, G. Chen, and X. Yao, "Multi-start JADE with knowledge transfer for numerical optimization," in *IEEE Congress on Evolutionary Computation*, Hefei, 2009, pp. 1889 - 1895.

[75] 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 , April 2011.

[76] M. G Epitropakis and et. al., "Enhancing Differential Evolution Utilizing Proximity-based Mutation Operators," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 99–119, 2011.

[77] M. Weber, *Parallel Global Optimization: Structuring Populations in Differential Evolution*. PhD Thesis, University of Jyvaskyla , 2010.

[78] T. Podoba, L. Tomsu, K. Vlcek, and M. Heczko, "Surface reconstrucion by means of AI," in *International Conference on Computational & Experimental Engineering and Sciences*, 2010, pp. 111-121.

[79] D. Davendra and et. el., "Chaos Driven Evolutionary Algorithm for the Traveling Salesman Problem," in *Traveling Salesman Problem, Theory and Applications*, D. Davendra, Ed. Zlin, Czech: InTech, 2010, ch. 4, pp. 55-70.

[80] B. V. Babu and S. A. Munawar, "Differential evolution strategies for optimal design of shell-and-tube heat exchangers," *Chemical Engineering Science*, vol. 62, pp. 3720 – 3739, 2007.

[81] G. C. Onwubolu and D. Davendra, *Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization*, J. Kacprzyk, Ed.: Springer, 2009.

[82] D. Davendra, I. Zelinka, and G. Onwubolu, "Hybrid Differential Evolution – Scatter Search Algorithm for Permutative Optimization," in *Evolutionary Computation*, W. P. D. Santos, Ed. Vienna, Austria, 2009, ch. 15, pp. 273-296.

[83] I. Triguero, S. Garcia, and F. Herrera, "Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification," *Pattern Recognition*, vol. 44, pp. 901–916, 2011.

[84] I. Triguero, S. Garcia, and F. Herrera, "A preliminary study on the use of differential evolution for adjusting the position of examples in nearest neighbor classification," IEEE, 2010.

[85] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization," *IEEE Transaction on Evolutionary Computing*, vol. 13, no. 2, pp. 398-417, April 2009.

[86] R. Mallipeddi and et. al., "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, pp. 1679–1696, 2011.

[87] A. Hiendro, "Multiple Switching Patterns for SHEPWM Inverters Using Differential Evolution Algorithms," *International Journal of Power Electronics and Drive System*, vol. 1, no. 2, pp. 94-103, December 2011.

[88] D. Davendra and G. Onwubolu, "Enhanced differential evolution hybrid scatter search for discrete optimization," in *IEEE Congress on Evolutionary Computation*, 2007, pp. 1156 - 1162.

[89] 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.

[90] J. Adeyemo and F. Otieno, "Optimizing planting areas using differential evolution (DE) and linear programming (LP)," *International Journal of Physical Sciences*, vol. 4, no. 4, pp. 212-220, April 2009.

[91] A. Chandra and S. Chattopadhyay, "Novel Approach of Designing Multiplier-less Finite Impulse Response Filter using Differential Evolution Algorithm," *Internation Journal of Intelligent Systems and Applications*, vol. 4, pp. 54-62, April 2012.

[92] D. Davendra and G. Onwubolu, "Flow shop scheduling using enhanced differential evolution algorithm," in *21st European Conference on Modelling and Simulation*, Orsoni, 2007, pp. 1-6.

[93] M. J. Reddy and D. N. Kumar, "Multiobjective Differential Evolution with Application to Reservoir System Optimization," *Journal of computing in civil engineering* , vol. 21, no. 2, pp. 136-146, 2007.

[94] M. Swain, R. Panda, and S. Tibrewal, "Study of Differential Evolutionary Algorithm in Blind Source Separation," *International Journal of Computer Applications*, vol. 30, no. 7, pp. 48-55, September 2011.

[95] J. J. Valdes, "Evolutionary Computation Based Nonlinear Transformations to Low Dimensional Spaces for Sensor Data Fusion and Visual Data Mining," in *IEEE Congress on Evolutionary Computation* , Ottawa, 2010 , pp. 1 - 8.

[96] S. Z. Zhao, P. N. Suganthan, and S. Das, "Self-adaptive Differential Evolution with Multi-trajectory Search for Large Scale Optimization," *Soft Computing*, vol. 15, no. 11, pp. 2175-2185, November 2011.

[97] M. Pant, R. Thangaraj, and V. P. Singh, "Optimization of Mechanical Design Problems Using Improved Differential Evolution Algorithm," *International Journal of Recent Trends in Engineering*, vol. 1, no. 5, pp. 21-24, May 2009.

[98] S. Almeida-Luz and et. al., "Applying Differential Evolution to the Reporting Cells Problem," in *International Multiconference on Computer Science and Information Technology*, 2008, pp. 65 – 71.

[99] R. Mendes and A. S. Mohais, "DynDE: a Differential Evolution for Dynamic Optimization Problems," in *IEEE Congress on Evolutionary Computing*, 2005, pp. 2808 - 2815.

[100] Y. Ao and H. Chi, "Dynamic Differential Evolution for Constrained Real-Parameter Optimization," *Journal of Advances in Information Technology*, vol. 1, no. 1, February 2010.

[101] X. Zhang and X. Jiang, "A minimax fitting algorithm for ultra-precision aspheric surfaces," in *International Conference on Metrology and Properties of Engineering Surfaces*, 2011, pp. 1-6.

[102] Y. Ao and H. Chi, "Differential Evolution Using Opposite Point for Global Numerical Optimization," *Journal of Intelligent Learning Systems and Applications*, vol. 4, pp. 1-19, 2012.

[103] Z. Yang, K. Tang, and X. Yao, "Self-adaptive Differential Evolution with Neighborhood Search," in *IEEE Congress on Evolutionary Computation*, 2008., pp. 1110-1116.

[104] H. P. Thethi, B. Majhi, and G. Panda, "Efficient system identification using a low complexity nonlinear network with differential evolution and its variant based training schemes ," *International Journal of Computer Applications*, vol. 31, no. 8, pp. 38-46, October 2011.

[105] B. Y. Qu and P. N. Suganthan, "Novel multimodal problems and differential evolution with ensemble of restricted tournament selection," in *IEEE Congress on Evolutionary Computation*, 2010, pp. 1 - 7.

[106] Y. Ao and H. Chi, "An Adaptive Differential Evolution Algorithm to Solve Constrained Optimization Problems in Engineering Design," *Engineering*, vol. 2, no. 1, pp. 65-77, 2010.

[107] A. Mandal, S. Das, and A. Abraham, "A differential evolution based Memetic Algorithm for workload optimization in power generation plants," in *11th International Conference on Hybrid Intelligent Systems*, Kolkata, 2011, pp. 271 - 276.

[108] G. T. S. Oliveira and S. F. P. Saramago, "A Contribution to the Study About Differential Evolution ," *Ciencia & Engenharia*, vol. 16, no. 1/2, pp. 1-8, January 2007.

[109] S. K. Goudos, Z. D. Zaharis, and T. V. Yioultsis, "Application of a differential evolution algorithm with strategy adaptation to the design of multi-band microwave filters for wireless communications ," *Progress In Electromagnetics Research*, vol. 109, pp. 123-137, 2010.

[110] A. Fialho, M. Schoenauer, and M. Sebag, "Fitness-AUC bandit adaptive strategy selection vs. the probability matching one within differential evolution: an empirical comparison on the bbob-2010 noiseless testbed," in *12th annual conference companion on Genetic and evolutionary computation*, Oregon, 2010, pp. 1-8.

[111] A. Fialho, W. Gong, and Z. Cai, "Probability Matching-based Adaptive Strategy Selection vs. Uniform Strategy Selection within Differential Evolution," in *12th annual conference companion on Genetic and evolutionary computation* , Oregon, 2010, pp. 1-7.

[112] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Differential Evolution with Multiple Strategies for Solving CEC2011 Real-world Numerical Optimization Problems," in *IEEE Conference*, 2011, pp. 1041-1048.

[113] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Multi-operator based evolutionary algorithms for solving constrained optimization problems," *Computers & Operations Research*, vol. 38, pp. 1877-1896, 2011.

[114] D. Zaharie, "Control of population diversity and adaptation in differential evolution algorithms," in *In proceeding of MENDEL*, vol. 9, 2003, pp. 41-46.

[115] J. Liu and J. Lampinen, "A Fuzzy Adaptive Differential Evolution Algorithm," *Soft Computing*, vol. 9, pp. 448-462, 2005.

[116] 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.

[117] 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.

[118] 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.

[119] 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.

[120] 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.

[121] X. Li. and M. Yin, "Modified differential evolution with self-adaptive parameters method," *Journal of Combinatorial Optimization*, vol. 31, no. 2, pp. 546-576, 2016.

[122] S. Guo and et al., "Improving Differential Evolution With a Successful-Parent-Selecting Framework," *Evolutionary Computation*, vol. 19, no. 5, pp. 717-730, 2015.

[123] S. Nama, A. K. Saha, and S. Ghosh, "A new ensemble algorithm of differential evolution and backtracking search optimization," *Industrial Engineering Computations*, vol. 7, no. 2, pp. 323-338, 2016.

[124] C. Brown and et al., "μJADE: adaptive differential evolution with a small population," *Soft Computing*, pp. 1-10, 2015.

[125] C. Segura and et al., "On the adaptation of the mutation scale factor in differential evolution," *Optimization Letters*, vol. 9, no. 1, 2015.

[126] J. Gou and et al., "Adaptive differential evolution with directional strategy and cloud model," *Applied Intelligence*, vol. 42, pp. 369–388, 2015.

[127] K. N. Kozlov, A. M. Samsonov, and M. G. Samsonova, "The differential evolution entirely parallel method for model adaptation in systems biology," *Biophysics*, vol. 60, no. 6, pp. 1016-1017, 2015.

[128] X. Dong and et al., "Enhancing local search of differential evolution algorithm for high dimensional optimization problem," in *Control Conference*, 2015, pp. 8411-8415.

[129] Y. Li and et al., "Differential evolution with an evolution path: A DEEP evolutionary algorithm," *IEEE TRANSACTIONS ON CYBERNETICS*, vol. 45, no. 9, pp. 1798-1810, 2015.

[130] L. Tang, Y. Dong, and J. Liu, "Differential Evolution With an Individual-Dependent Mechanism," *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, vol. 19, no. 4, pp. 560-574, 2015.

[131] W. Gong and Z. Cai, "Differential evolution with ranking-based mutation operators," *IEEE TRANSACTIONS ON CYBERNETICS*, vol. 43, no. 6, pp. 2066-2081, 2013.

[132] S. Biswas, S. Kundu, and S. Das, "Inducing niching behavior in differential evolution through local information sharing," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 246-263, 2015.

[133] S. Bandyopadhyay and A. Mukherjee, "An algorithm for many-objective optimization with reduced objective computations: A study in differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 400-413, 2015.

[134] W. Gao, G. G. Yen, and S. Liu, "A dual-population differential evolution with coevolution for constrained optimization.," *IEEE TRANSACTIONS ON CYBERNETICS*, vol. 45, no. 5, pp. 1108-1121, 2015.

[135] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "An improved self-adaptive differential evolution algorithm for optimization problems," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 89-99, 2013.

[136] C. L. Lu and et al., "Systematic identification of multiple tumor types in microarray data based on hybrid differential evolution algorithm," *Technology and Health Care*, vol. 24, no. 1, pp. 237-244, 2015.

[137] G. Saeidi, M. R. Moniri, and A. Shahzadi, "Self-adaptive Differential Evolution-Based Multiple Model Variable Rate Particle Filter for Trajectory Tracking," *Arabian Journal for Science and Engineering*, pp. 1-10, 2015.

[138] B. Gao, J. Zhu, and W. Lang, "Research on Improved Differential Evolution Algorithm based on Hybrid Multi-strategy and its Application," *International Journal of Hybrid Information Technology*, vol. 8, no. 4, pp. 147-156, 2015.

[139] B. P. De and et al., "Optimal design of high speed symmetric switching CMOS inverter using hybrid harmony search with differential evolution," *Soft Computing*, pp. 1-19, 2015.

[140] X. Li and M. Yin, "Parameter estimation for chaotic systems by hybrid differential evolution algorithm and artificial bee colony algorithm," *Nonlinear Dynamics*, vol. 77, no. 1-2, pp. 61-71, 2014.

[141] A. R. Yildiz, "Hybrid Taguchi-differential evolution algorithm for optimization of multi-pass," *Applied Soft Computing*, vol. 13, no. 3, pp. 1433-1439, 2013.

[142] I. J. Fister and et al., "Artificial neural network regression on ensemble strategies in differential evolution," in *International Conference on Soft Computing*, 2014, pp. 65-70.

[143] Y. Bazi ` and et al., "Differential evolution extreme learning machine for the classification of hyperspectral images," *Geoscience and Remote Sensing Letters*, vol. 11, no. 6, pp. 1066-1070, 2014.

[144] G. Wang and et al., "Hybrid krill herd algorithm with differential evolution for global numerical optimization," *Neural Computing and Applications*, vol. 25, no. 2, pp. 297-308, 2014.

[145] A. R. Yildiz, "A new hybrid differential evolution algorithm for the selection of optimal machining parameters in milling operations," *Applied Soft Computing*, vol. 13, no. 3, pp. 1561-1566, 2013.

[146] M. A. Zamee and et al., "Differential evolution algorithm based load frequency control in a two-area conventional and renewable energy based nonlinear power system," in *International Conference on the Development in the in Renewable Energy Technology*, 2016, pp. 1-6.

[147] R. Li and et al., "Competition Differential Evolution Strategy for low cross polarization synthesis of conformal array antenna," in *IEEE International Wireless Symposium* , 2015, pp. 1-4.

[148] H. Ma, C. Chen, and D. Dong, "Differential Evolution with Equally-Mixed Strategies for Robust Control of Open Quantum Systems," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2015, pp. 2055-2060.

[149] S. Sumithra and T. A. Victoire, "Differential Evolution Algorithm with Diversified Vicinity Operator for Optimal Routing and Clustering of Energy Efficient Wireless Sensor Networks," *The Scientific World Journal*, pp. 1-7, 2015.

[150] S. Kundu and et al., "A modified differential evolution-based combined routing and sleep scheduling scheme for lifetime maximization of wireless sensor networks," *Soft Computing*, vol. 19, no. 3, pp. 637-659, 2015.

[151] K. S. Reddy and S. K. Sahoo, "An approach for FIR filter coefficient optimization using differential evolution algorithm," *International Journal of Electronics and Communications*, vol. 69, no. 1, pp. 101-108, 2015.

[152] B. Mohanty, S. Panda, and P. K. Hota, "Controller parameters tuning of differential evolution algorithm and its application to load frequency control of multi-source power system," in *International Journal of Electrical Power & Energy Systems*, 2014, pp. 77-85.

[153] D. Jiang, "Study on Improved Differential Evolution Algorithm for Solving Complex Optimization Problem," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, no. 12, pp. 241-248, 2014.

[154] H. Salehinejad, S. Rahnamayan, and H. R. Tizhoosh, "Micro-Differential Evolution: Diversity Enhancement and Comparative Study," *Applied Soft Computing*, vol. arXiv:1512.07980, 2015.

[155] A. W. Mohamed, H. Z. Sabry, and M. Khorshid, "An alternative differential evolution algorithm for global optimization," *Journal of Advanced Research*, vol. 3, no. 2, pp. 149–165, April 2012.

[156] L. D. S Coelho and V. C. Mariani, "Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect," *IEEE Transactions on*

Power Systems, vol. 21, no. 2, pp. 989-996, May 2006.

[157] C. Lin and et al., "An Interactively Recurrent Functional Neural Fuzzy Network with Fuzzy Differential Evolution and Its Applications," *Sains Malaysiana*, vol. 44, no. 12, pp. 1721-1728, 2015.

[158] M. Yang and et al., "Differential evolution with auto-enhanced population diversity," *IEEE TRANSACTIONS ON CYBERNETICS45*, vol. 45, no. 2, pp. 302-315, 2015.

[159] S. Chen and M. Song, "DSP-based direct neural control for thrust active magnetic bearing learned through adaptive differential evolution," in *Automatic Control Conference*, 2015, pp. 96-101.

[160] 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.

[161] R. Mallipeddi and P. N. Suganthan, "Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies," in *Swarm, Evolutionary, and Memetic Computing*, Springer Berlin Heidelberg, 2010, pp. 71-78.

[162] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4-32, February 2011.

[163] B. L. Miller and D. E. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise," *Complex Systems*, vol. 9, no. 3, pp. 193-212, 1995.

[164] T. Blickle and L. Thiele, "A Comparison of Selection Schemes used in Genetic Algorithms," TIK Institut fur Technische und Kommunikationsnetze, Swiss Federal Institute of Technology, Zurich,Switzerland, TIK-Report 11, Dec. 1995.

[165] H. Takagi and D. Pallez, "Paired Comparisons-based Interactive Differential Evolution," in *World Congress on Nature & Biologically Inspired Computing*, Coimbatore, 2009, pp. 475 - 480.

[166] P. Posik and J. Kubalik, "Experimental Comparison of Six Population-Based Algorithms for Continuous Black Box Optimization," *Evolutionary Computation*, vol. 20, no. 4, pp. 483-508, 2012.

[167] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential Evolution Using a Neighborhood-based Mutation Operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526-553, 2009.

[168] M.G. Epitropakis, V. P. Plagianakos, and M. N. Vrahatis, "Balancing the exploration and exploitation capabilities of the Differential Evolution Algorithm," in *IEEE World Congress on Computational Intelligence*, Hong Kong, 2008, pp. 2686 - 2693.

[169] 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.

[170] M. Jamil and X. Yang, "A literature survey of benchmark functions for global optimisation problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150-194, 2013.

[171] R. Landa, Y. Rojas, and G. Toscano-Pulido, "An approach for estimating separability and its application on high dimensional optimization," in *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, Philadelphia, 2012, pp. 951-958.

[172] P. Caamano and et al., "Are evolutionary algorithm competitions characterizing landscapes appropriately," in *Proceedings of the 13th annual conference companion on Genetic and*

evolutionary computation, Dublin, 2011, pp. 695-702.

[173] N. Noman and H. Iba, "Accelerating Differential Evolution Using an Adaptive Local Search," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 107-125, February 2008.

[174] M. M. Ali and A. Torn, "Population set-based global optimization algorithms: Some modifications and numerical studies," *Computer and Operatation Research*, vol. 31, no. 10, pp. 1703–1725, 2004.

[175] A.P. Piotrowski and J. J. Napiorkowski, "The grouping differential evolution algoriothm for multi-dimensional optimization problems," *Control and Cybernetics*, vol. 39, no. 2, pp. 527-540, 2010.

[176] X. Dong and et al., "A comparative study of differential evolution and genetic algorithms for optimizing the design ofwater distribution systems," *Applied Physics & Engineering*, vol. 13, no. 9, pp. 674-686, 2012.

[177] Q. Abbas, J. Ahmad, and H. Jabeen, "A Novel Tournament Selection based Differential Evolution Variant for Continuous Optimization Problems," *Mathematical Problems in Engineering*, vol. Article ID 205709, doi:10.1155/2015/205709, pp. 1-21, 2015.

[178] 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.

[179] Y. Zhang and et al., "Binary Structuring Elements Decomposition Based on an Improved Recursive Dilation-Union Model and RSAPSO Method," *Mathematical Problems in Engineering*, pp. 1-12, 2014.

[180] Y. Yoon and H. Kim, *The Roles of Crossover and Mutation in Real-Coded Genetic Algorithms, Bio-Inspired Computational Algorithms and Their Applications*, Shangce Gao, Ed., 2012.

[181] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization: An Overview," *Swarm Intelligence*, vol. 1, no. 1, pp. 33-57, 2007.

[182] X. L. Dong, S. Q. Liu, T. Tao, S. P. Li, and K. L. Xin, "A comparative study of differential evolution and genetic algorithms for optimizing the design of water distribution systems," *Journal of Zhejiang University SCIENCE A*, vol. 13, no. 9, pp. 674-686, 2012.

[183] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64-79, 2008.

[184] A. W. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution," in *In AI 2004: Advances in artificial intelligence*, Springer Berlin Heidelberg, 2005, pp. 861-872.

[185] A. P. Engelbrecht, *Computational intelligence an introduction*, 2nd ed. England: John Wiley & Sons, 2007.

[186] J. Kacprzyk, *Studies in Computational Intelligence, Volume 155*, H. R. Tizhoosh and M. Ventresca, Eds.: Springer, 2008.

[187] 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.

[188] J. Ronkkonen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential," in *IEEE Congress on Evolutionary Computation*, 2005, pp. 506-513.

[189] C. O. Godfrey and D. Donald, *Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization*, J. Kacprzyk, Ed.: Springer, Heidelberg, 2009.

[190] H.N.A Hamed and et al., "Multi-objective K-means evolving spiking neural network model based

on differential evolution," in *International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering*, 2015, pp. 379-383.

[191] A. O. Ibrahim and et al., "Intelligent multi-objective classifier for breast cancer diagnosis based on multilayer perceptron neural network and Differential Evolution," in *International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering*, 2015, pp. 422-427.

Appendix

1). $f_{1:}$

Name: Sphere model

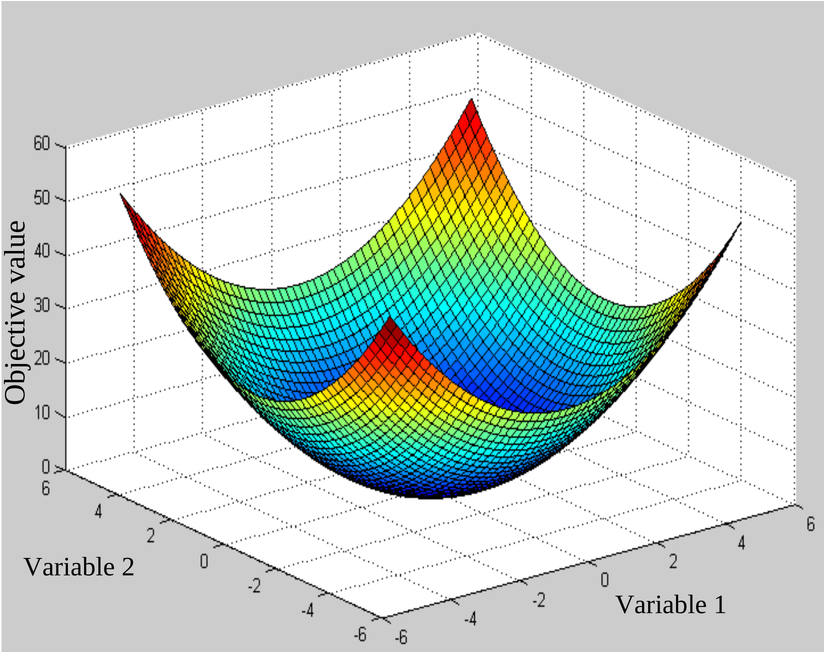
Equation: $f(x)=\sum_{i=0}^{n} x_i^2$

Search Space: $-5.12 \leq x_i \leq 5.12$

Characteristics:

- Separable,
- Multimodal

Optima: 0



A.1 Sphere model function landscape of two variables

2). $f_{2:}$

Name: Axis parallel hyperellipsoid

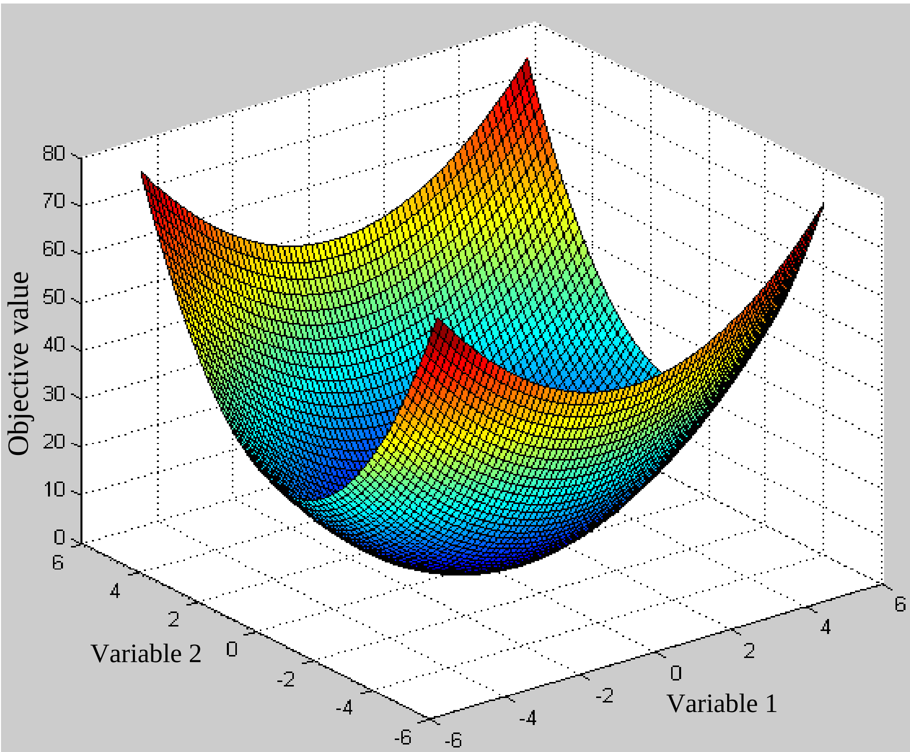
Equation: $f(x)=\sum_{i=0}^{n} i \cdot x_i^2$

Search Space: $-5.12 \leq x_i \leq 5.12$

Characteristics:

- Separable
- Unimodal

Optima: 0



A.2 Axis parallel hyperellipsoid landscape of two variables

3). *f3:*

Name: Schwefel's problem 1.2

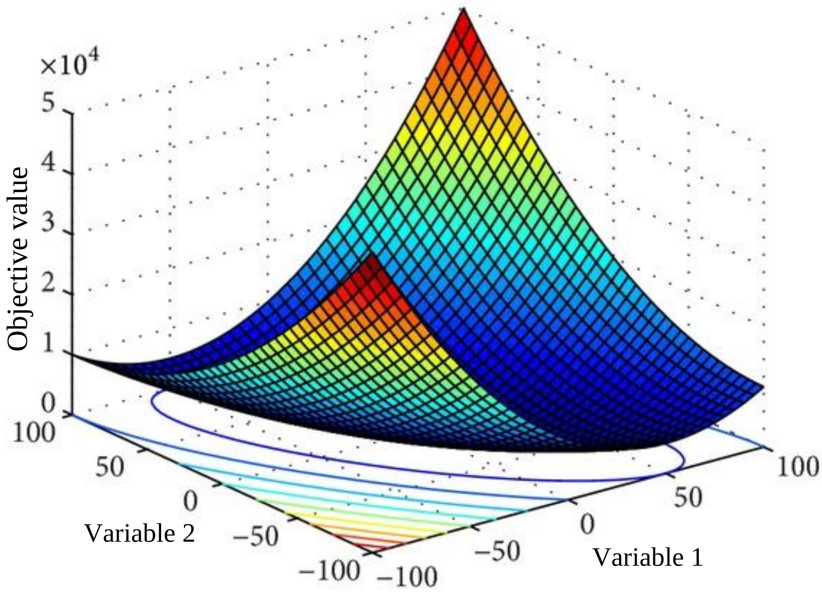
Equation: $f(x) = \sum_{i=0}^{n} \left(\sum_{j=0}^{n} x_j \right)^2$

Search Space: $-65 \le x_j \le 65$

Characteristics:

- Non-separable
- Unimodal

Optima: 0



A.3 Schwefel's problem 1.2 landscape of two variables

4). *f4:*

Name: Rosenbrock's valley

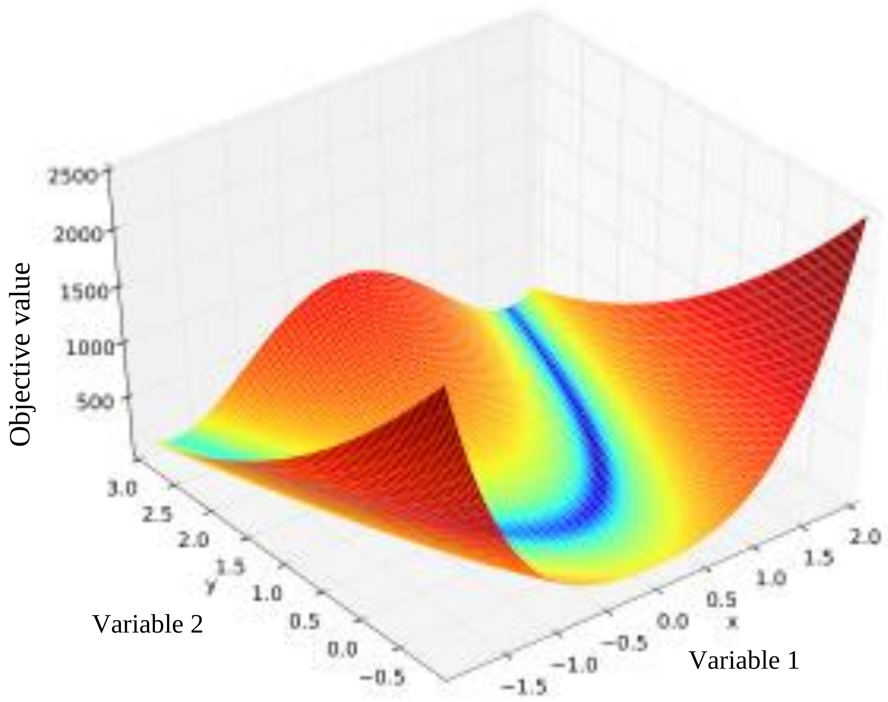
Equation: $f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$

Search Space: $-30 \le x_i \le 30$

Characteristics:

- Non-separable
- Unimodal

Optima: 0



A.4 Rosenbrock's valley landscape of two variables

5). $f_{5:}$

Name: Rastrigin's function

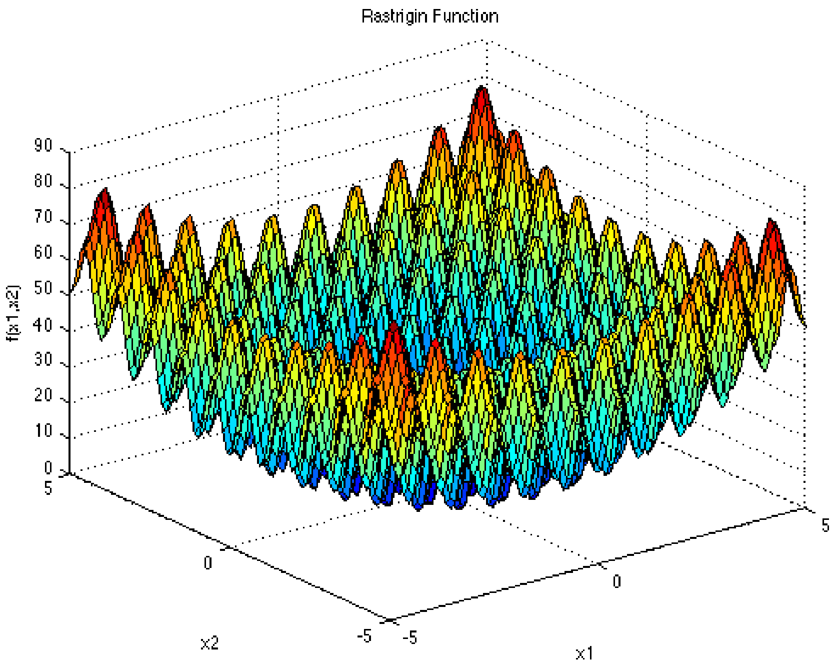
Equation: $f(x) = 10n + \sum^{n} (x_i^2 - 10\cos(2\pi x_i))$

Search Space: -5.12

Characteristics:

- Separable
- Multimodal

Optima: 0



A.5 Rastrigin's function landscape of two variables

6). $f_{6:}$

Name: Griewangk's function

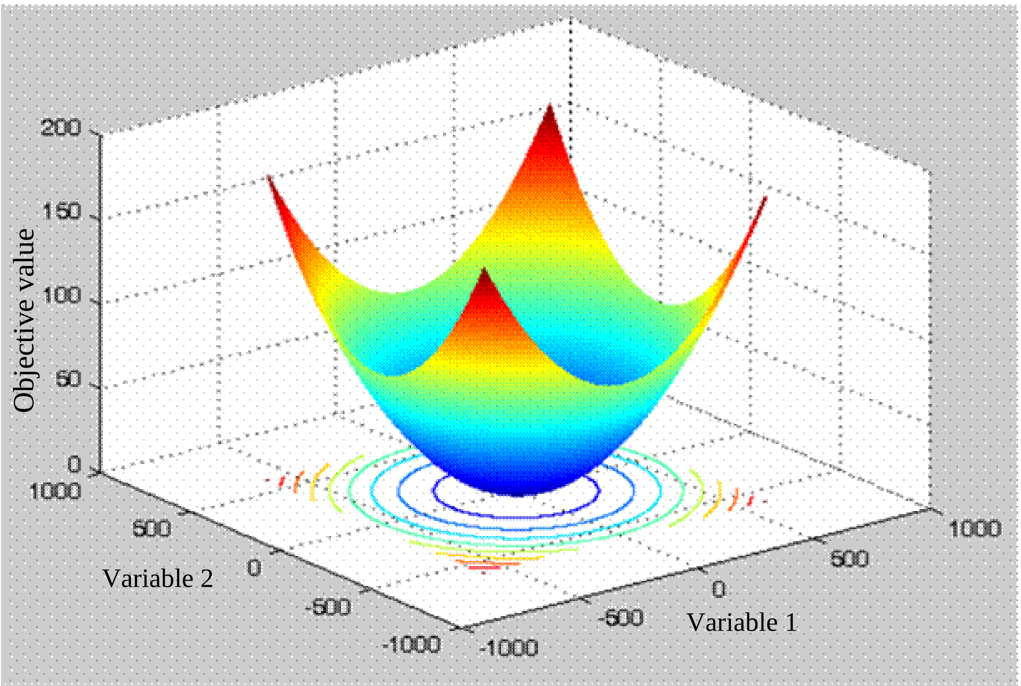
Equation: $f(x) = \sum_{i=1}^{n} (\frac{x_i^2}{4000} - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) + 1$

Search Space: -600 $\leq x_i \leq$ 600

Characteristics:

- Non-separable
- Multimodal

Optima: 0



A.6 Griewangk's function landscape of two variables

7). *f7:*

Name: Sum of different power

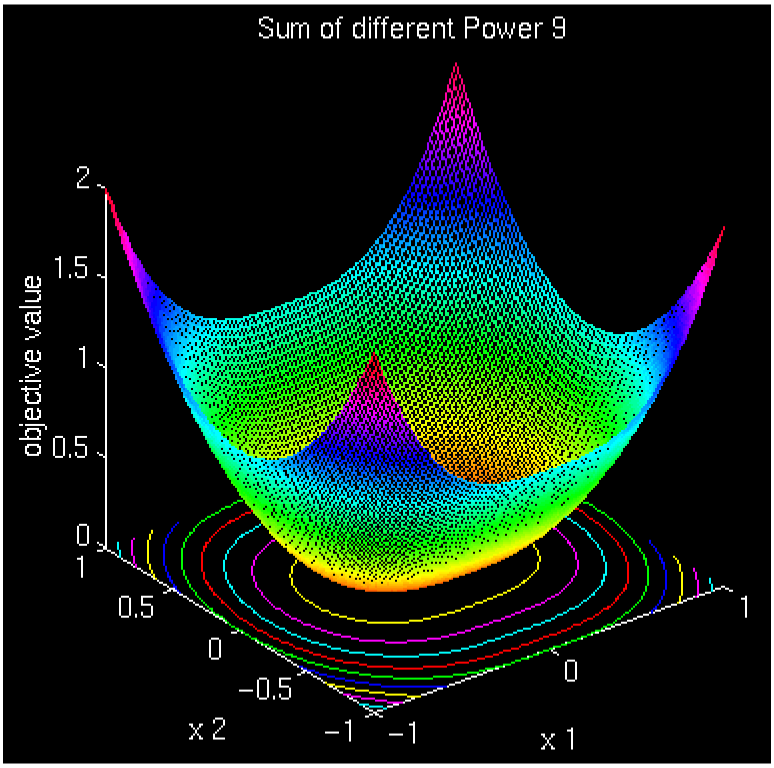
Equation: $f(x) = \sum_{i=1}^{n} |x_i|^{(i+1)}$

Search Space: $-1 \leq x_i \leq 1$

Characteristics:

- Non-separable
- Multimodal

Optima: 0



A.7 Sum of different power landscape of two variables

8). *f8:*

Name: Ackley's path function

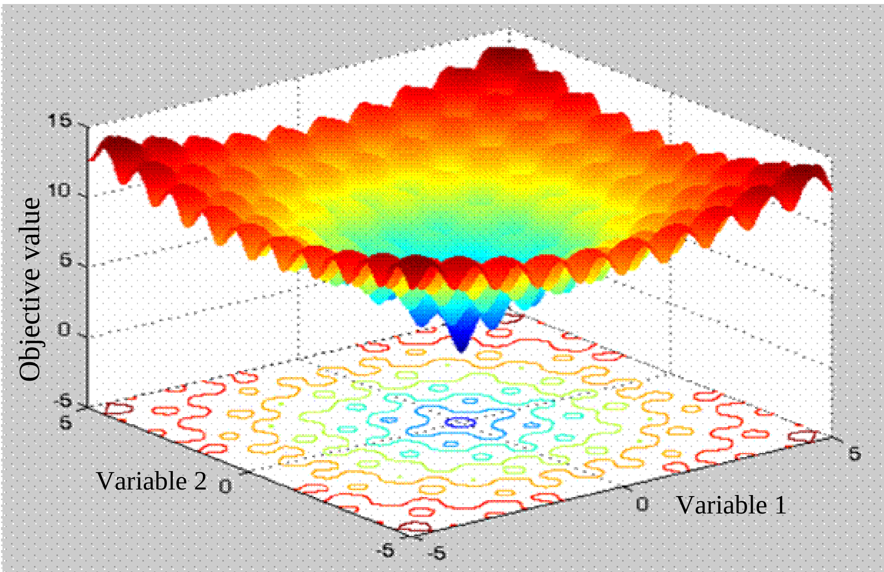
Equation: $$f(x) = -20\exp\left(-0.2\sqrt{\frac{\sum_{i=1}^{n} x_i^{2}}{n}}\right) - \exp\left(\frac{\sum_{i=i}^{n} Cos(2\pi x_i}{n}\right) + 20 + e$$

Search Space: $-32 \leq x_i \leq 32$

Characteristics:

- Non-separable
- Multimodal

Optima: 0



A.8 Ackley's path function landscape of two variables

9). $f_{9:}$

Name: Levy function

Equation:

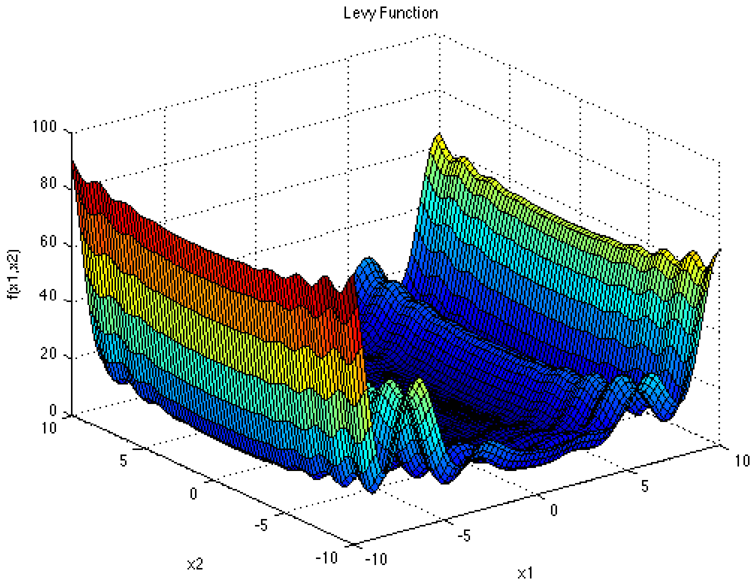
$$f(x)=0.1\left[\sin^2(3\pi x_1)+\sum_{i=1}^{n-1}(x_i-1)^2\times\left(1+\sin^2(3\pi x_i+1)\right)+(x_n-1)\left(1+\sin^2(2\pi x_n)\right)\right]$$

Search Space: $-10\leq x_i\leq 10$

Characteristics:

- Separable
- Multimodal

Optima: 0



A.9 Levy function landscape of two variables

10). $f_{10:}$

Name: Zakharov function

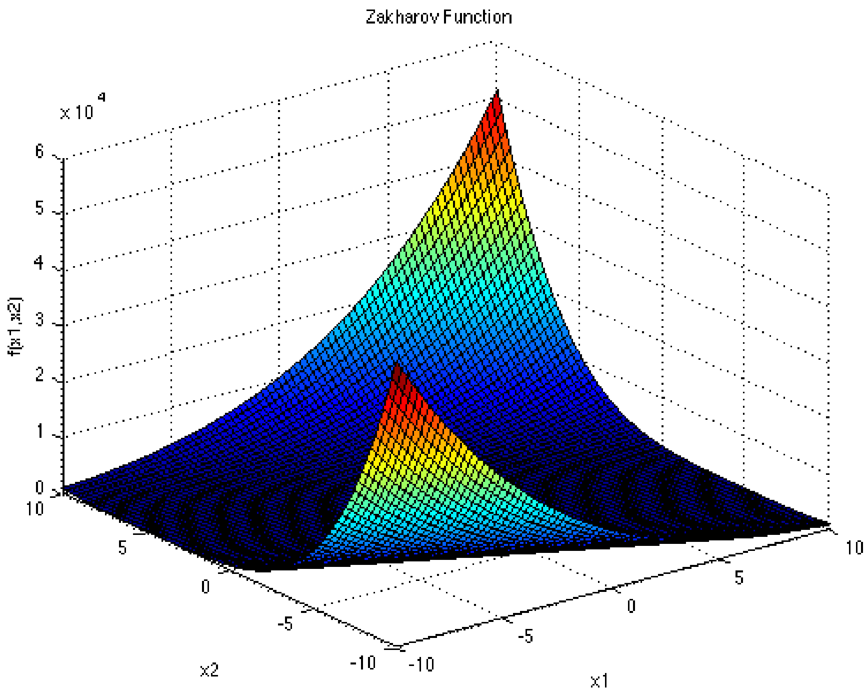
Equation: $$f(x)=\sum_{i=1}^{n}x_i^2+\left(\sum_{i=i}^{n}0.5ix_i\right)^2+\left(\sum_{i=i}^{n}0.5ix_i\right)^4$$

Search Space: $-5\leq x_i\leq 10$

Characteristics:

- Non-separable
- Multimodal

Optima: 0



A.10 Zakharov function landscape of two variables

11). *f11:*

Name: Schwefel's problem 2.22

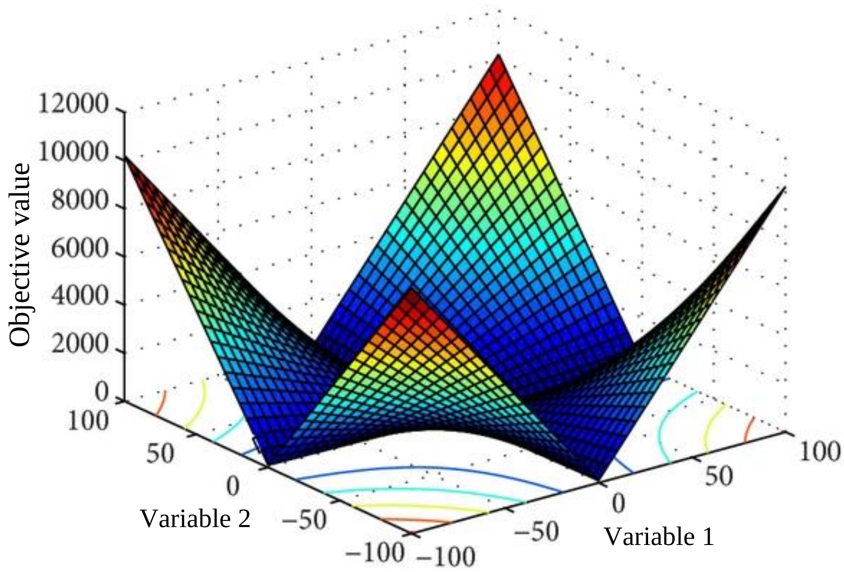
Equation: $$f(x)=\sum_{i=1}^{n}|x_i|+\prod_{i=1}^{n}|x_i|$$

Search Space: $-10 \le x_i \le 10$

Characteristics:

- Non-separable
- Unimodal

Optima: 0



A.11 Schwefel's problem 2.22 landscape of two variables

12). *f12:*

Name: Step function

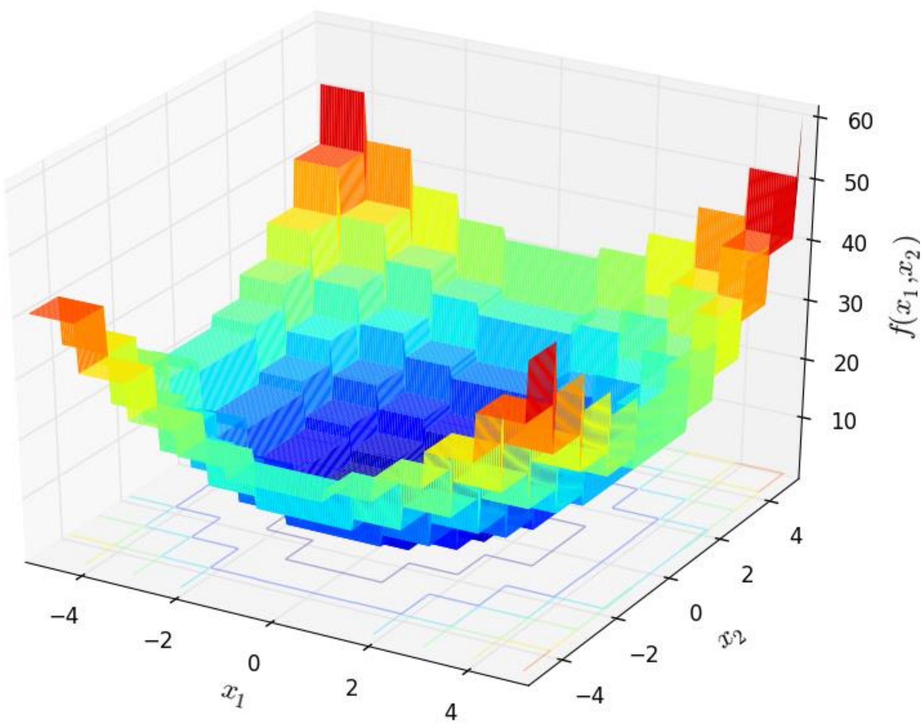
Equation: $$f(x)=\sum_{i=1}^{n}\left(\lfloor x_i+0.5\rfloor\right)^2$$

Search Space: $-100 \le x_i \le 100$

Characteristics:

- Separable
- Unimodal

Optima: 0



A.12 Step function landscape of two variables

13). $f_{13:}$

Name: De Jong's function 4 (no noise)

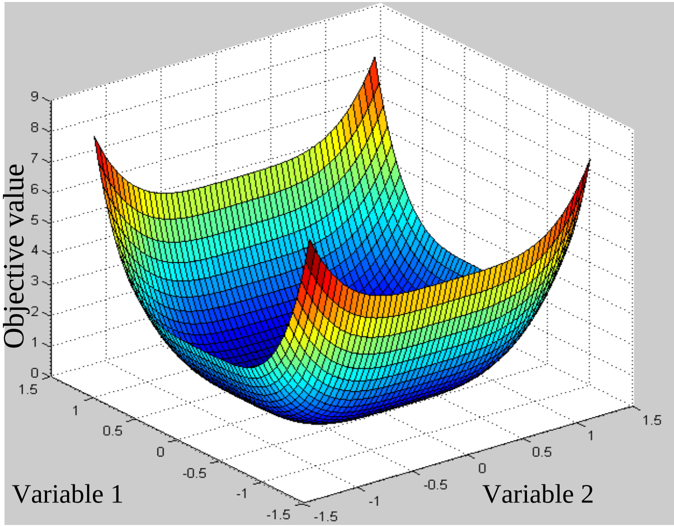
Equation: $f(x)=\sum_{i=1}^{n} i x_i^{4}$

Search Space: $-1.28 \le x_i \le 1.28$

Characteristics:

- Separable
- Unimodal

Optima: 0



A.13 De Jong's function 4 landscape of two variables

14). $f_{14:}$

Name: Alpine function

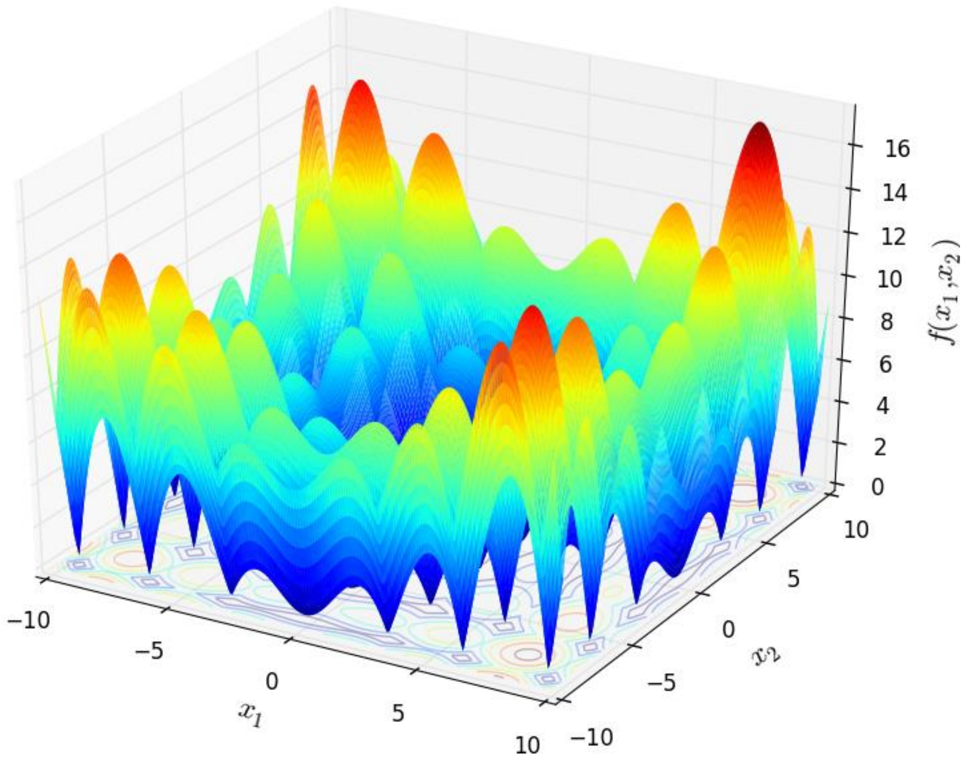
Equation: $f(x)=\sum_{i=1}^{n} \left| x_i \sin(x_i) + 0.1 x_i \right|$

Search Space: $-10 \le x_i \le 10$

Characteristics:

- Separable
- Multimodal

Optima: 0



A.14 Alpine function landscape of two variables

15). f_{15}:

Name: Levy and Montalvo Problem

Equation: $$f(x)=\left(\frac{\pi}{n}\right)\left(10\sin^2(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin 2(\pi y_{i+1})\right]\right)+(y_n-1)^2$$

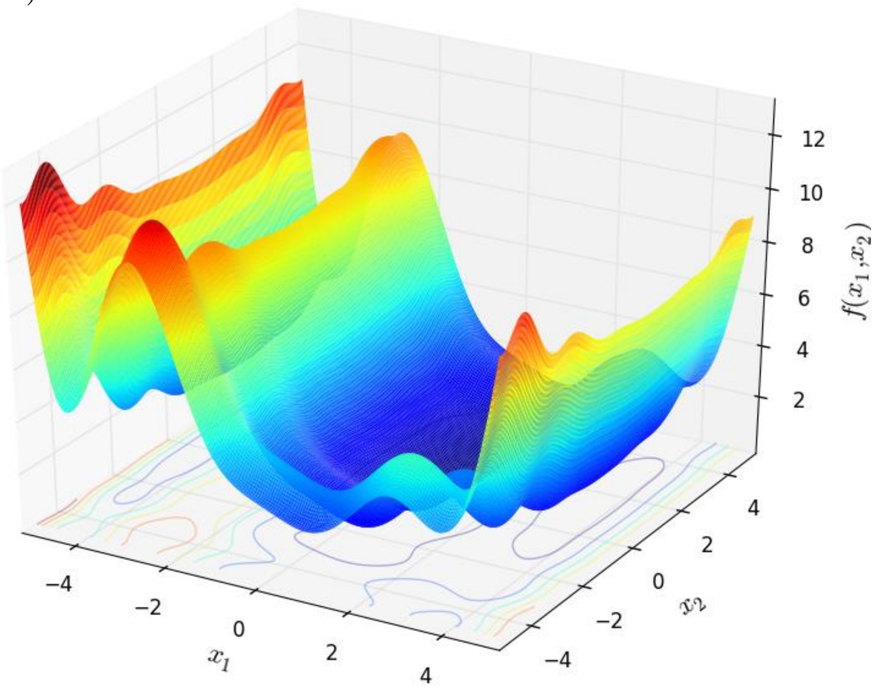
Where $y_i=1+\frac{1}{4}(x_i+1)$

Search Space: $-10\leq x_i\leq 10$

Characteristics:

- Separable
- Multimodal

Optima: 0



A.15 Levy and Montalvo Problem landscape of two variables

16). f_{16}:

Name: Neumaier 2 Problem

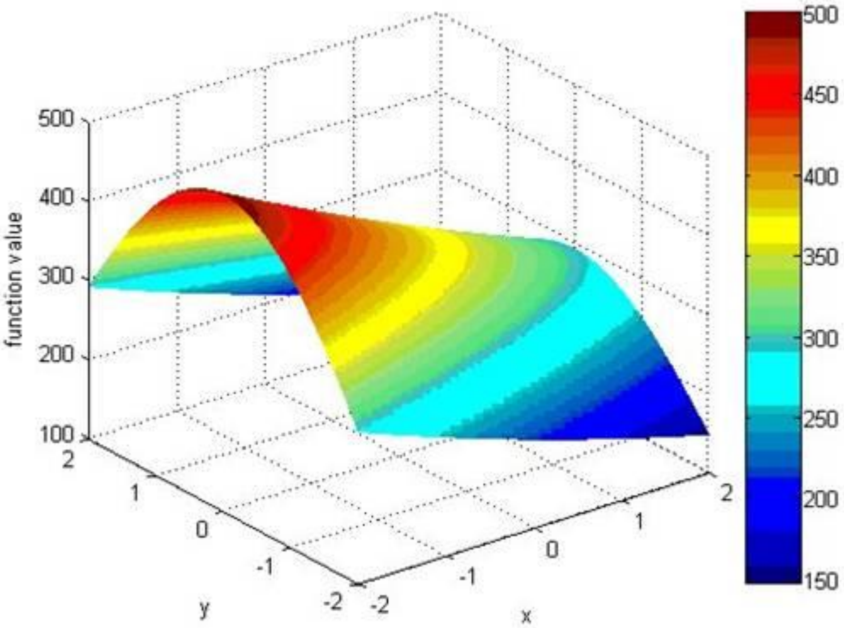
Equation: $$f(x)=\sum_{i=1}^{n}(x_i-1)^2-\sum_{i=2}^{n}(x_i x_{i-1})$$

Search Space: $-n^2\leq x_i\leq n^2$

Characteristics:

- Separable
- Unimodal

Optima: 0



A.16 Neumaier 2 Problem landscape of two variables

17). f_{17}:

Name: Cosine Mixture

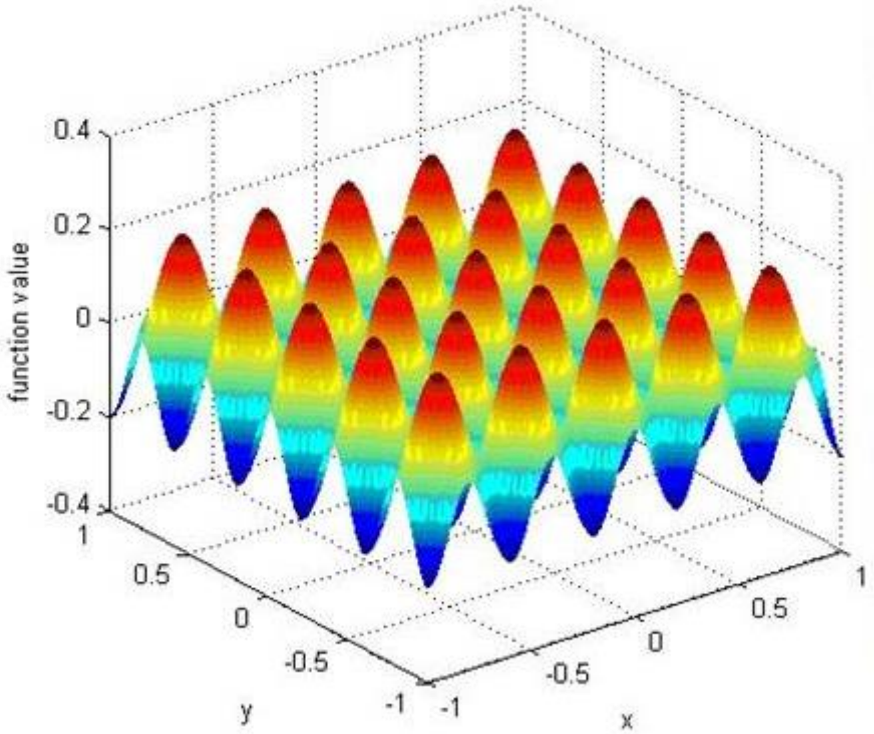
Equation: $f(x) = -0.1\sum_{i=1}^{n}\cos(5\pi x_i) + \sum_{i=1}^{n} x_i^2 \qquad \|x\| = \sqrt{\sum_{i=1}^{n} x_i^2}$

Search Space: $-1 \le x_i \le 1$

Characteristics:

- Separable
- Multimodal

Optima: *-0.1×(n)*



A.17 Cosine Mixture landscape of two variables

18). f_{18}:

Name: Cigar

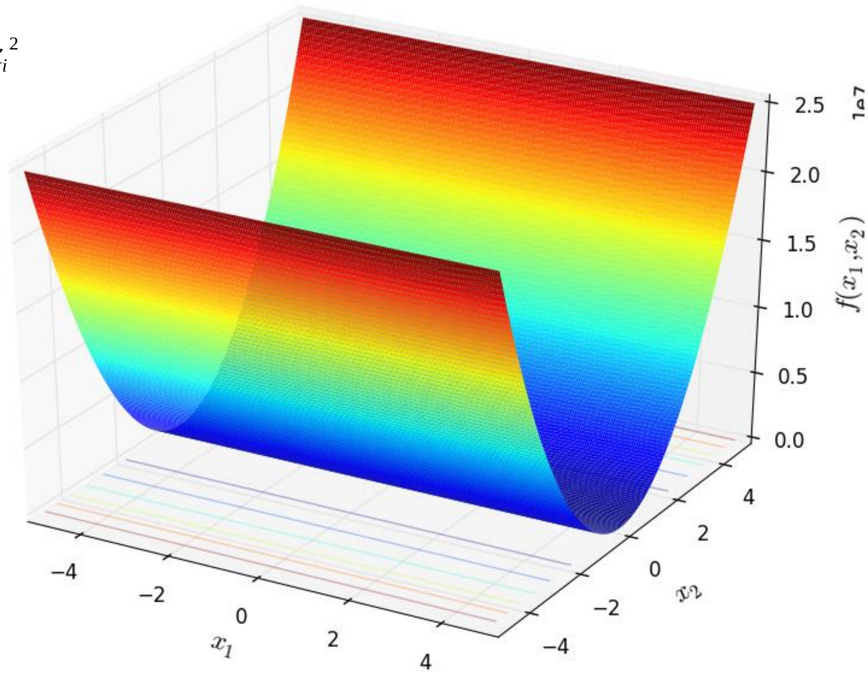
Equation: $f(x) = x_1^2 + 100000\sum_{i=1}^{n} x_i^2$

Search Space: $-10 \le x_i \le 10$

Characteristics:

- Separable
- Multimodal

Optima: 0



A.18 Cigar function landscape of two variables

19). f_{19}:

Name: Function ‘15’

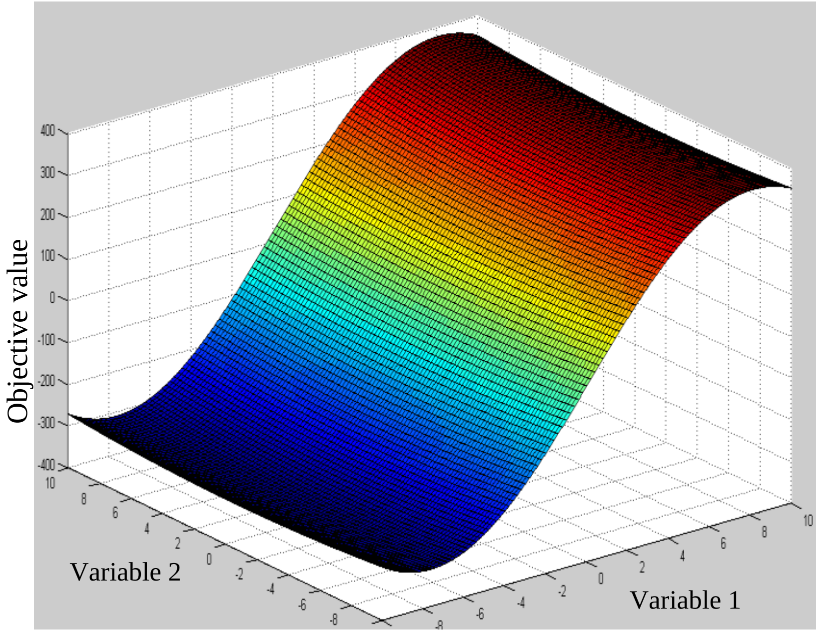
Equation: $f(x)=\sum_{i=1}^{n-1}\left[0.2x_i^{\ 2}+0.1x_i^{\ 2}\sin(2x_i)\right]$

Search Space: $-10 \le x_i \le 10$

Characteristics:

- Separable
- Multimodal

Optima: 0



A.19 Function ‘15’ landscape of two variables

20). f_{20}:

Name: Ellipse Function

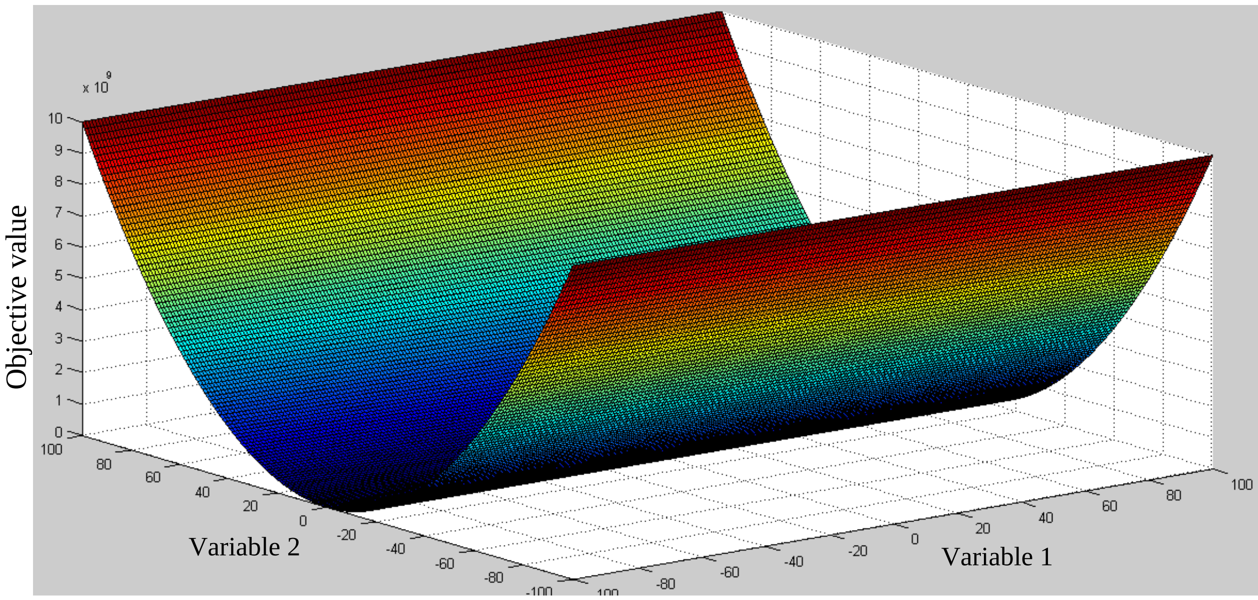
Equation: $f(x)=\sum_{i=1}^{n}(10^{6(\frac{i-1}{n-1})}.x_i^2)$

Search Space: $-100 \le x_i \le 100$

Characteristics:

- Separable
- Unimodal

Optima: 0



A.20 Ellipse Function landscape of two variables

21). *f21:*

Name: Tablet Function

Equation: $f(x) = 10^4 x_1^2 + \sum_{i=2}^{n} x_i^2$

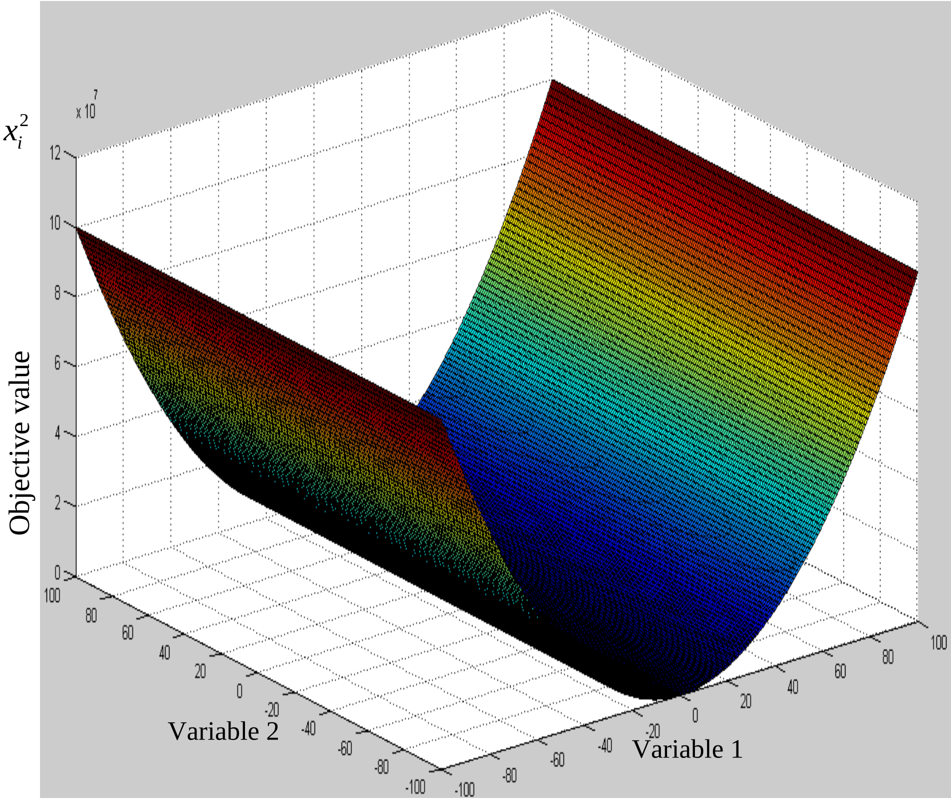
Search Space: $-100 \le x_i \le 100$

Characteristics:

- Separable
- Unimodal

Optima: 0

2D-Graph(in Matlab)



A.21 Tablet Function landscape of two variables

22). *f22:*

Name: Schewel function

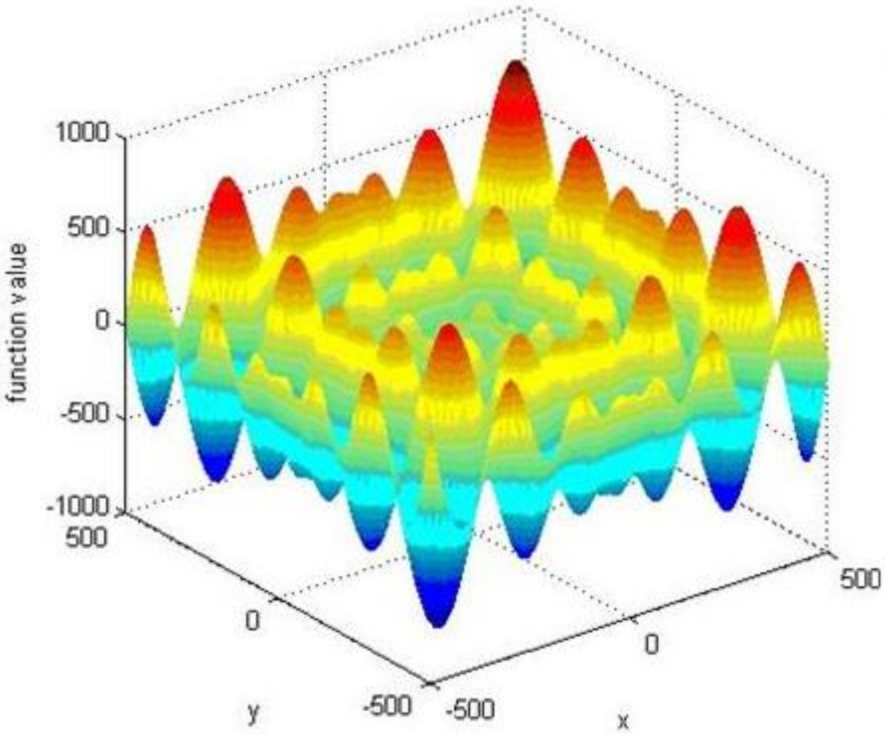
Equation: $f(x) = \sum_{i=1}^{n} ((x_1 - x_i^2)^2 + (x_i - 1)^2)$

Search Space: $-32 \le x_i \le 32$

Characteristics:

- Separable
- Multimodal

Optima: 0



A.22 Schewel function landscape of two variables

23). f_{23}:

Name: Deflected Corrugated Spring

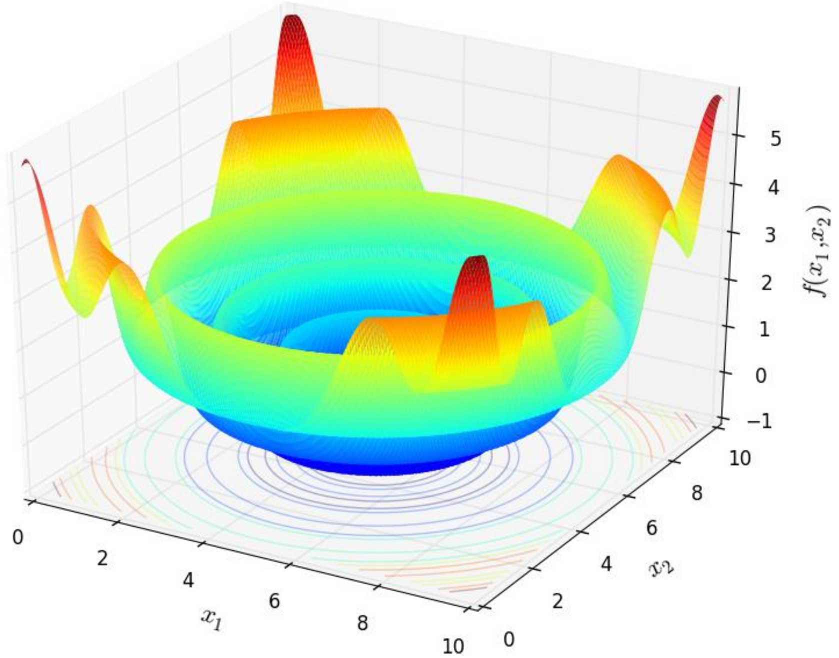
Equation: $$f(x) = 0.1\sum_{i=1}^{n}\left((x_i-\alpha)^2 - \cos\left(K\sqrt{\sum_{i=1}^{n}((x_i-\alpha)^2)}\right)\right)$$

Search Space: $0 \le x_i \le 10 \quad K=5$
$\alpha = 5 \quad x_i \in [0, 2\alpha]$

Characteristics:

- Separable
- Multimodal

Optima: 0



A.23 Deflected Corrugated Spring landscape of two variables

24). f_{24}:

Name: Mishra 1 global optimization problem

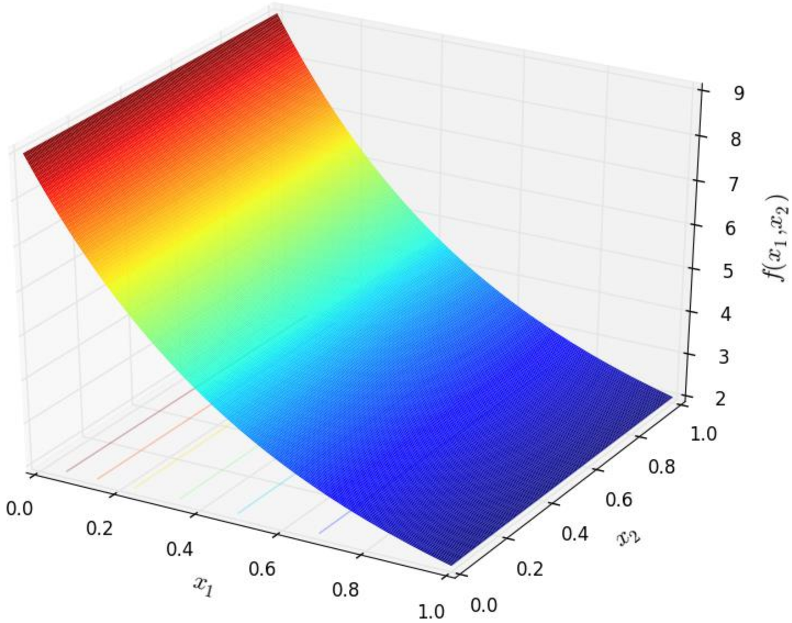
Equation: $$f(x) = (1+x_n)^{x_n} \quad \text{where} \quad x_n = n - \sum_{i=1}^{n-1} x_i$$

Search Space: $0 \le x_i \le 1$

Characteristics:

- Non-separable
- Multimodal

Optima: 2



A.24 Mishra 1 global optimization problem landscape of two variables

25). f_{25}:

Name: Mishra 2 global optimization problem

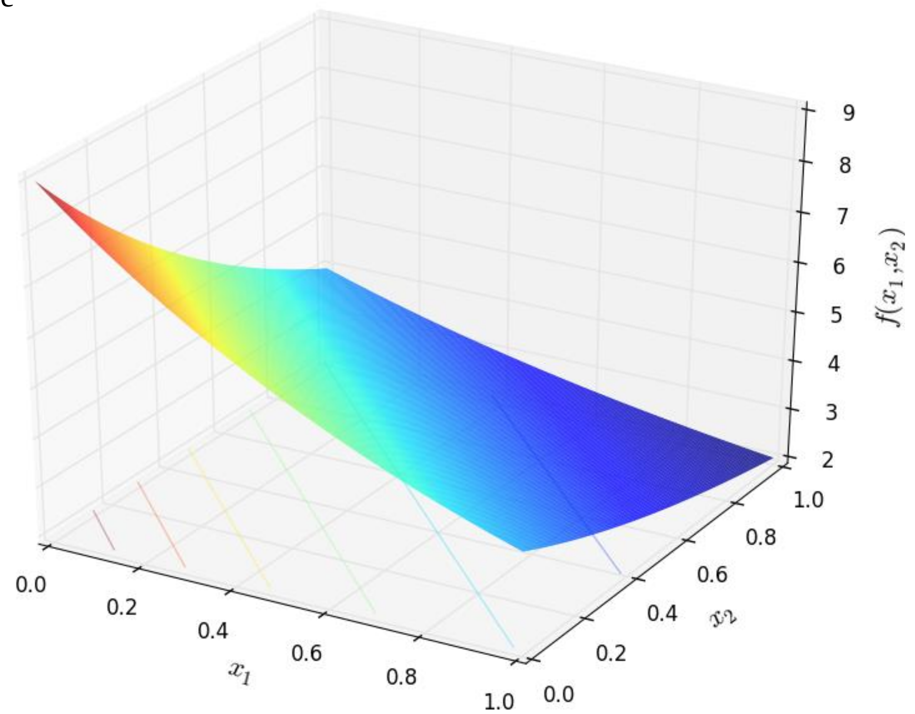
Equation: $f(x) = (1 + x_n)^{x_n}$ where $x_n = n - \sum_{i=1}^{n-1} \frac{(x_i + x_{i+1})}{2}$

Search Space: $0 \le x_i \le 1$

Characteristics:

- Non-separable
- Multimodal

Optima: 2



A.25 Mishra 2 global optimization problem landscape of two variables

26). f_{26}:

Name: MultiModal global optimization problem

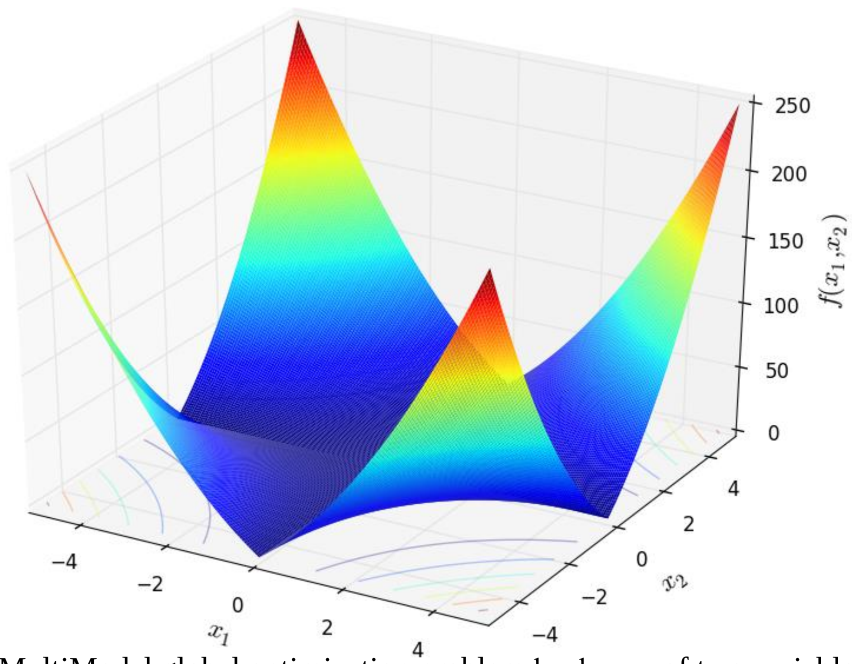
Equation: $f(x) = \left(\sum_{i=1}^{n} |x_i|\right)\left(\prod_{i=1}^{n} |x_i|\right)$

Search Space: $-10 \le x_i \le 10$

Characteristics:

- Separable
- Multimodal

Optima: 0



A.26 MultiModal global optimization problem landscape of two variables

27). f_{27}:

Name: Quintic global optimization problem

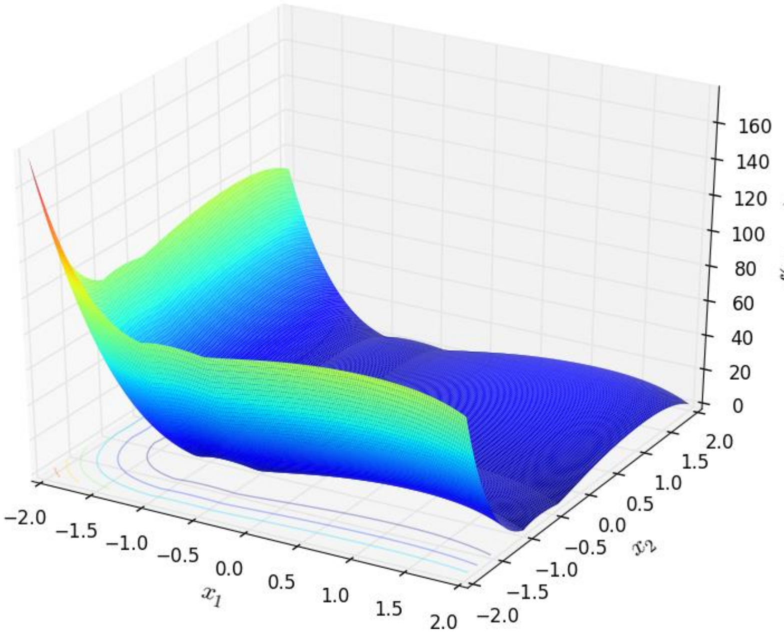
Equation: $f(x) = \sum_{i=1}^{n} \left| x_i^5 - 3x_i^4 + 4x_i^3 + 2x_i^2 - 10x_i - 4 \right|$

Search Space: $-10 \le x_i \le 10$

Characteristics:

- Separable
- Multimodal

Optima: -1



A.27 Quintic global optimization problem landscape of two variables

28). f_{28}:

Name: Stochastic global optimization problem

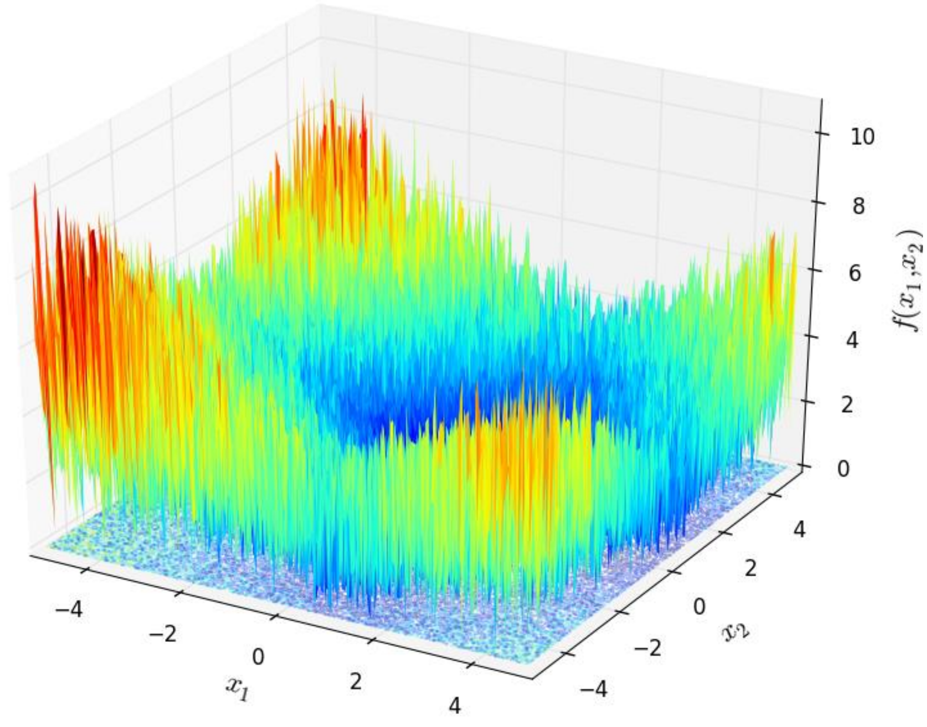
Equation: $f(x) = \sum_{i=1}^{n} \varepsilon_i \left| x_i - \frac{1}{i} \right|$

Search Space: $-5 \le x_i \le 5$

Characteristics:

- Separable
- Multimodal

Optima: 0



A.28 Stochastic global optimization problem landscape of two variables

29). f_{29}:

Name: Stretched V global optimization problem

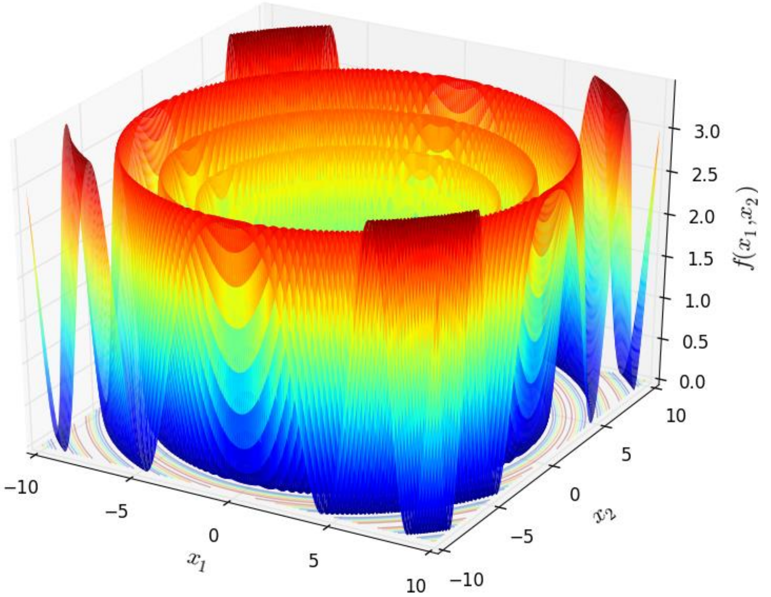
Equation: $$f(x)=\sum_{i=1}^{n-1} t^{1/4}\left[\sin\left(50t^{0.1}\right)+1\right]^2 \quad \text{where} \quad t=x_{i+1}^2+x_i^2$$

Search Space: $-10 \le x_i \le 10$

Characteristics:

- Non-separable
- Multimodal

Optima: 0



A.29 Stretched V global optimization problem landscape of two variables

30). f_{30}:

Name: XinSheYang function

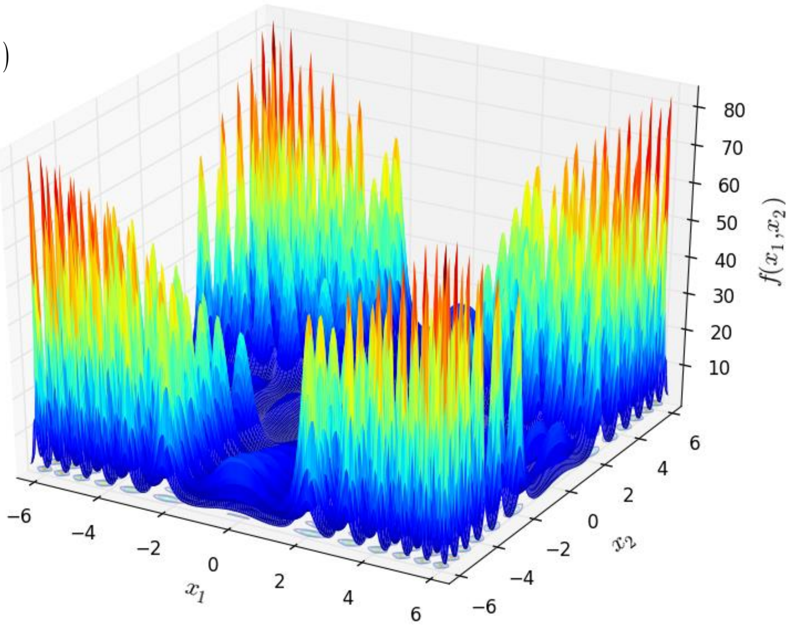
Equation: $$f(x)=\left(\sum_{i=1}^{n}|x_i|\right)\Big/ e^{\sum_{i=1}^{n}\sin\left(x_i^2\right)}$$

Search Space: $-2\pi \le x_i \le 2\pi$

Characteristics:

- Non-separable
- Multimodal

Optima: 0



A.30 XinSheYang function landscape of two variables