INTELLIGENT AUTOMATION & SOFT COMPUTING, 2017
http://dx.doi.org/10.1080/10798587.2017.1295678

Random Controlled Pool base Differential Evolution Algorithm (RCPDE)

Qamar Abbas[a], Jamil Ahmad[b] and Hajira Jabeen[a]

[a]Computer Department, Iqra University, Islamabad, Pakistan; [b]Computer Department, Abasyn University, Islamabad, Pakistan

ABSTRACT
This paper presents a novel random controlled pool base differential evolution algorithm (RCPDE) where powerful mutation strategy and control parameter pools have been used. The mutation strategy pool contains mutations strategies having diverse parameter values, whereas the control parameter pool contains varying nature pairs of control parameter values. It has also been observed that with the addition of rarely used control parameter values in these pools are highly beneficial to enhance the performance of the DE algorithm. The proposed mutation strategy and control parameter pools improve the solution quality and the convergence speed of DE algorithm. The simulation results of the proposed RCPDE algorithm shows significant performance as compared to other algorithms when tested over a set of multi-dimensional benchmark functions.

KEYWORDS
Differential evolution; pool; strategy; mutation; control parameters

1. Introduction

Evolutionary algorithms (EA) mimic the biological evolutionary process of a stochastic search for an optimal solution. Various EA algorithms are successfully applied to various optimization applications (Huang, He, & Yang, 2013; Velagic & Osmic, 2013). EA adapts genetic, inherent and survival of the fittest in finding the optimal solution of a given problem (Engelbrecht, 2007). Various operators such as mutation, crossover and selection are used to generate a new solution for optimum value in the specified search space (Brest, Greiner, Boskovic, Mernik, & Zumer, 2006). Differential evolution (DE), proposed by Storn and Price (1995) is a stochastic population based evolutionary algorithm, which offers many advantages over other evolutionary algorithms such as ease of use, better speed and greater probability of finding the global optima for function optimization (Brest, Greiner, Boskovic, Mernik, & Zumer, 2006; Price, Storn, & Lampinen, 2005). DE has been successfully applied to various real world problems, for example; electrical power systems (Yuan, Wang, Zhang, & Yuan, 2009), microwave engineering (Chowdhury et al., 2010), robotics (Smirnov & Jastrzebski, 2009), Bioinformatics (Marchiori, Moore, & Rajapakse, 2007), chemical engineering (Hao, Chen, Wu, & Yu, 2004), pattern recognition (Maulik & Saha, 2009), artificial neural networks (Dragoi, Curteanu, Galaction, & Cascaval, 2013) and signal processing (Liu, Li, & He, 2010).

DE is considered to be a population based algorithm, where a population of potential solutions is randomly initialized within solution search space. All potential solutions are equally likely to be selected as the parent. The candidate solutions evolve, overtime, by exploring the entire search space to locate the optima of the objective function (Yao, Liu, & Lin, 1999). New vector is generated by adding the weighted difference between two population vectors to a third vector at each iteration of the DE algorithm (Oliveira, 2007). Three vectors randomly selected from the existing population are used to generate a new vector. It is evident from the literature that DE has shown better performance for numerical benchmark optimization, when compared to Genetic Algorithm and Particle Swarm Optimization (Das, Abraham, & Konar, 2008; Xu & Li, 2007). There are many parameters in the DE algorithm like Population "NP", mutation probability "F" and Crossover "CR". The DE algorithm mutation variants are formed by the linear combination of existing population members. The trial vector and target vector forms the mutant vector in DE. Throughout this paper x_i denotes the target vector (or current vector); u_i represents the trial vector and v_i as a mutant vector. In the DE algorithm, different mutation schemes are used to create the trial vector by using any combination of current, best and random vectors. The behavior of the DE algorithm is influenced by the selection of mutation strategy and crossover scheme along with mutation probability "F" and Crossover rate "CR" (Das, Abraham, Chakraborty, & Konar, 2009; Storn & K, 1997) as their control parameters.DE mutation strategies can be formed by the combinations of current vector, random vector(s), better vector and best vector. The order, number and name of vector(s) are very important considerations in this regard. DE mutation strategies are generated by using a combination of random and/or best vector(s). In random selection, best and worst members have the same probability of selection as a parent. The worst parent may lead to the worst child. A novel variant of DE is proposed in this research that will be helpful in avoiding the selection of bad performing individuals. The proposed variant favors selection individuals with better fitness that will be helpful in enhancing the convergence speed and searching capability of DE algorithm. The rest of the paper is organized in the following manner. Literature survey is presented in Section-II. Various crossover schemes are given in Section-III of the paper. Section-IV contains the proposed DE variant. The detail of benchmark functions is given in Section-V. Finally, Section-VI presents results and discussion.

CONTACT Qamar Abbas qamar.bhk@gmail.com

2. Literature survey

Zaharie has introduced the population diversity mechanism using parameter adaption in his paper (Zaharie, 2003). The Multi-population concept is used to bring equilibrium in exploration and exploitation in DE algorithm. A Fuzzy controller is introduced by Liu and Lampinen (2005) to manage the values of DE control parameters F and CR. They have used parameter vector change measures for control parameter F and CR. To determine the control parameter F and CR values, a new self adaptive version of DE algorithm (jDE) is introduced by Brest, Greiner, Boskovic, Mernik, and Zumer (2006). 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 $\tau_1 = 0.1$, $\tau_2 = 0.1$, respectively. A new parameter adaption method in introduced as an adaptive DE with optional archive by Zhang and Sanderson (2009). 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 control parameters 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 the best vector. An optional archive is used that contains an inferior solution and then by using this concept a new population is generated from the union of the current population and archive.

The self-adaption of the control parameter and self-adaptive strategy selection of DE algorithm are proposed by Qin, Huang, and Suganthan (2009). They have used self adaption of the 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 the 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, Suganthan, Pan, and Tasgetiren (2011) have used a 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 a binomial and 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 is assigned to each vector to generate its corresponding target vector. In the global local version of the DE algorithm introduced by Das et al. (2009) 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, the donor vector is generated by combining local and global neighbourhoods by using α and β, which are scaling factors respectively. In the local neighbourhood, two random vectors are generated and the best from their neighbourhood is selected, while in the global neighbourhood, a best of the 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 Minhazul Islam, Das, Ghosh, Roy, and Suganthan (2012). New mutation strategy utilizes the concept of q% best population and selection 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,* p top-ranked individual 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, Cai, Ling, and Li (2011). 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 an archive,"DE/current-to-pbest" with archive, "DE/rand-to-pbest" without archive to form a strategy pool. A pool of three convention mutation and crossover strategies is used in composite DE (CoDE) by Wang, Cai, and Zhand (2011). 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 the DE algorithm in introduced by Li and Yin (2016). They have used two probabilistic rules to balance the exploration and exploitation in the DE algorithm. The two rules are based on 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 compared it with the other state of the art algorithms. Guo et.al has incorporated the concept of successful-parent-selection framework in the DE algorithm (Guo, Yang, Hsu, & Tsai, 2015). They have used "achieve of successful" solutions and then select parents from that "achieve of successful" after, when a solution is not updated for some certain period of time. They have compared their proposed method with a 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 (2016). 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, Jin, Leach, and Hodgson (2015) have introduced a small population based concept in the adaptive differential evolution. They have a population size of 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 was introduced by Segura, Coello, Segredo, and Leon (2015).

3. DE Algorithm

DE algorithm has three different parameters; a population of size NP, crossover control parameter CR and difference vector amplification parameter F. Each population member in DE is represented as a D-dimensional parameter vector. In the DE algorithm, population is initialized randomly that is supposed to cover the entire search space. Each vector in the DE is represented by $x_{i,G}$, where $i = 1, 2, 3,\ldots\ldots$, NP and G is generation number. New offsprings in the DE algorithm are generated by mutation, crossover and selection operators. Three different donor names, trial and target vectors are used in the DE algorithm for various vectors, where a donor vector is created in the mutation operation, trial vector is created in the crossover operation and target vector is the current vector of population. The details of these operators with index $i = 1, 2, 3...NP$, $j = 1, 2, 3....D$ is as follows:

Mutation: In a mutation operation, the mutant vector also called donor vector is created. Donor vector $v_{i,G+1}$ of ith population member is calculated by adding the weighted difference of two vectors to third vector.

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

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

Crossover: DE crossover strategies control the number of inherited components from the mutant vector to form a target vector. Binomial and Exponential are main crossover schemes (Ali, Pant, & Abraham, 2009; Mezura-Montes, Reyes, & Coello Coello, 2006; Storn & K, 1997). The DE crossover rate parameter (CR) influences the size of perturbation of the base (target) vector to ensure the population diversity (Buhry, Giremus, Grivel, Saïghi, & Renaud, 2009; Das et al., 2009). Following are the binomial and exponential crossover schemes.

Binomial Crossover: In the crossover operation of the DE algorithm, a trial vector is formed. In the 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:

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

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

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:

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

where $i = 1, 2, 3...NP$, $j = 1, 2, 3....D$ and $\langle\ \rangle_D$ 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 $u_{i,G}$.

Selection: In the DE algorithm, new population members are formed using the selection operation. Selection operator uses the greedy approach by comparing fitness of the 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. The following equation is used for selection operator:

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

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

4. Proposed Random Controlled Pool base Differential Evolution Algorithm (RCPDE)

Various researchers have investigated control parameters and DE mutation strategies in the last decade to fine tune the evolutionary process for faster convergence and gain the prior knowledge about convergence and obtain prior knowledge. The prior knowledge is helpful in designing a control parameter pool and strategy pool of the DE algorithm. In the proposed RCPDE algorithm, we have used five mutation strategies for the strategy candidate pool and four control parameters to form a control parameter candidate pool adopted from (Abbas, Ahmad, & Jabeen, 2015; Dong, Liu, Tao, Li, & Xin, 2012; Rahnamayan, Tizhoosh, & Salama, 2008; Wang et al., 2011). The varying behavior of the control parameters of the parameter pool and strategies of the strategy pool are found helpful in solving different kinds of problems. These pools are discussed in detail in the following sections:

Strategy Pool

(1) DE/rand to best/1: This mutation strategy utilizes information of best solutions, which provides fast convergence (Mallipeddi & Suganthan, 2010). It uses two difference vectors and perturbs random vectors in the direction of best vector that may incorporate more diversity by producing more trial vectors (Mallipeddi & Suganthan, 2010; Qin et al., 2009; Wang et al., 2011). This mutation strategy proves itself to be one of the better performing mutation strategies (Abbas et al., 2015).

(2) DE/rand/1: It bears stronger exploration in the DE algorithm that is helpful to incorporate diversity in the population (Qin et al., 2009).

(3) DE/current to rand/1: It is used to solve the rotated problems more effectively than other strategies (Iorio & Li, 2005; Qin et al., 2009).

(4) DE /rand/2: It improves diversity and perturbs random vector. It uses two difference vectors that incorporates more diversity than using a single difference vector by producing more trial vectors (Mallipeddi & Suganthan, 2010; Qin et al., 2009; Zhang & Sanderson, 2009).

(5) TSDE/bin: It improves the convergence speed of the DE algorithm and helps to escape the local optima problem (Abbas et al., 2015).

Control Parameter Pool

Crossover generates new solutions by shuffling competing vectors information between population individuals. It increases the population diversity and the opportunities to reproduce superior individuals in the current population (Engelbrecht, 2007; Kacprzyk, 2008). Small value of CR is useful to solve separable problems, while the large value of CR helps to solve multimodal problems (Price et al., 2005; Ronkkonen, Kukkonen, & Price, 2005). New information in the population is introduced by the mutation operation that generates random variations in population individuals (Godfrey & Donald, 2009). Small value of F is helpful for exploitation, while the large value of F is helpful to maintain Exploration in the DE algorithm (Qin et al., 2009). 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 the proposed RCPDE algorithm.

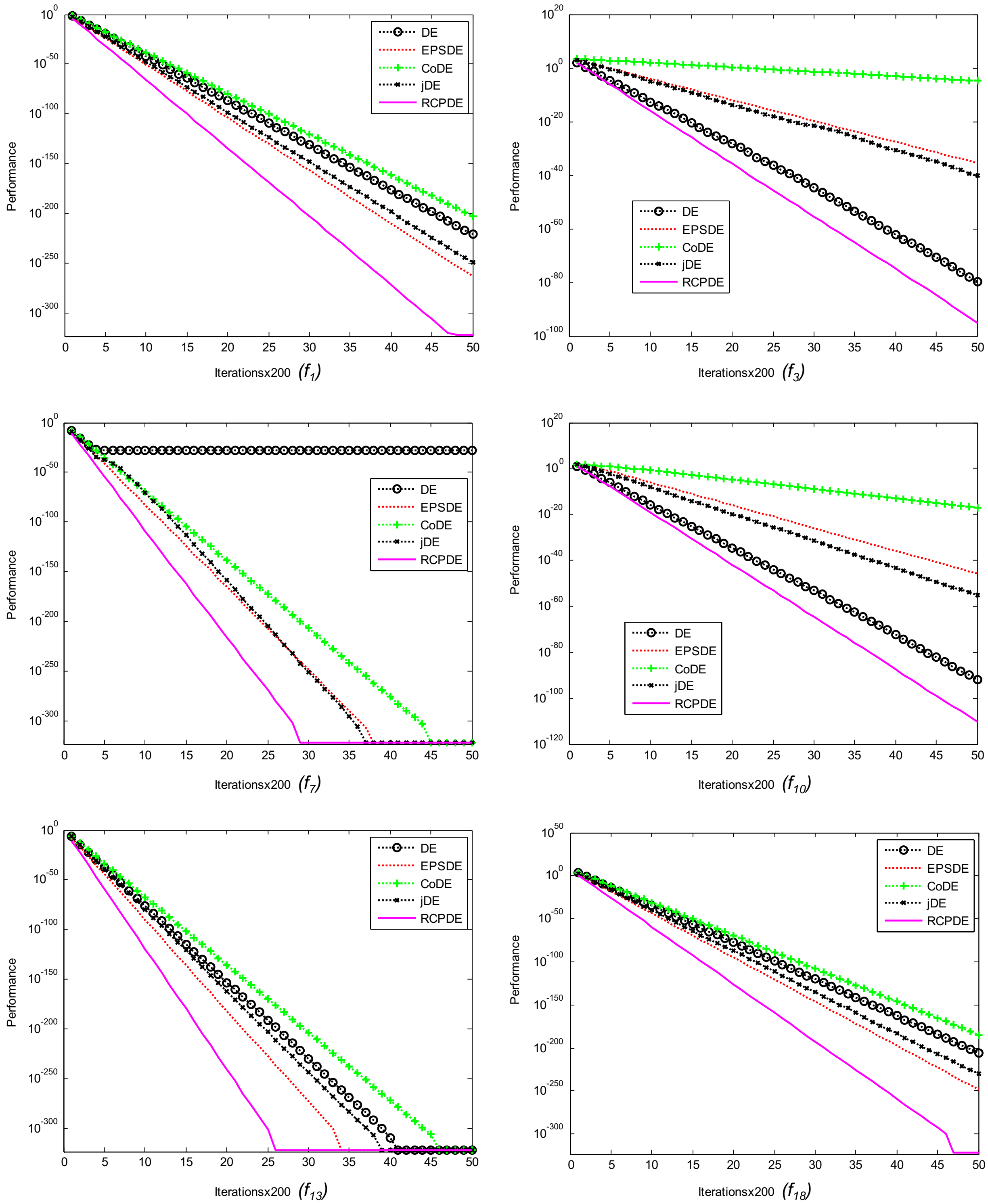


Figure 1. -20D Convergence graphs of some functions showing iterations horizontally and performance vertically.

(1) $F = 1.0$; CR = 0.1 (Wang et al., 2011)
(2) $F = 0.8$; CR = 0.2 (Wang et al., 2011)
(3) $F = 0.7$; CR = 0.5 (Abbas et al., 2015; Dong et al., 2012)
(4) $F = 0.5$; CR = 0.9 (Rahnamayan et al., 2008)

Figure 1 shows the pseudocode of the proposed RCPDE algorithm. The working of the proposed algorithm starts with the parameter pool and strategy candidate pool initialization. After the initialization of DE population numbers; the fitness value of each population member is calculated. Population members are evolved after selecting the mutation strategy and control parameter values. After the evolutionary process, the optimal solution is obtained. The proposed RCPDE algorithm is also implemented through computer simulation and results are presented in SectionVI of this paper.

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 $P_G = \{X_{1,GG}, \ldots\ldots X_{NP,G}\}$ for generation G = 0, randomly initialize each population member $X_{i,G} = \{x^1_{i,G} \ldots\ldots x^D_{i,G}\}$ where $i = 1 \ldots\ldots\ldots\ldots NP$

(2) Randomly initialize the control parameter memory from control parameter pool and strategy memory from strategy pool for each population member $X_{i,G} = \{x^1_{i,G} \ldots\ldots x^D_{i,G}\}$ where $i = 1 \ldots\ldots\ldots\ldots NP$.

(3) FOR i = 1 to NP

Calculate fitness $f(X_{i,G})$ for each population member $X_{i,G}$ using the parameter pool value and mutation strategy assigned in step-2.

END FOR

(4) WHILE the stopping criterion is not true

Step 4.1 *Vectors selection*

Select vectors to be used in the equation of mutation strategy ***S*** (given in equaions1–4) from current Population

Step 4.2 *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 ***Si*** from memory strategy and ith control parameter ***Fi*** from control parameter memory.

END FOR

Step 4.3 *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 the control parameter ***CRi*** from control parameter memory.

END FOR

Step 4.4 *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 $f(U_{i,G} \le f(X_{i,G})$, THEN $X_{i,G+1} = U_{i,G}$, $f(X_{i,G}) = f(U_{i,G})$

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

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 4.5 *Increment Generation Number* $G = G+1$

Step 5 *END WHILE*

Figure 1 *Pseudocode of Random Controlled Pool base Differential Evolution algorithm*

5. Parameter Study and Test Functions

A comprehensive set of N-dimensional test functions taken from (Abbas et al., 2015) having varying characteristics 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 Tables 1 and 2. Experimental results are generated by using control parameter population Size $N_p = 3D$ where D is the dimension and dimensions are used as 10D, 20D and 30D with iterations 5,000, 10,000 and 15,000, respectively. Average fitness value is calculated after 30 trials. Number of Function, called (NFC) is generated for maximum NFC 10^4 *DIM (Abbas et al., 2015). 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 functions.

6. Results and Discussion

Experimental results of NFC and average fitness value performance parameters are presented in this section. N-dimensional functions having varying characteristics are used to evaluate the performance of proposed RCPDE and its competitors. Table 3 and 4 represent simulation results of NFC performance parameters for 10D, 20D and 30D for benchmark functions. Average fitness is reported in Table 3 and Table 4 for the same set of benchmark functions. The results of the number of function calls and average fitness are generated over 30 independent trials. Results are generated using a parameter setting given in SectionV of this paper.

The experimental results of the NFC performance parameter are generated using the setting discussed in SectionV above. Because of large size of numerical values, the results are presented in multiple tables. Tables 3 and 4 contains NFC values and their corresponding standard deviation, where best values are reported as boldfaces. Experimental result of DE, EPSDE, CoDE, jDE and the proposed RCPDE are obtained using 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.* The DE Algorithm has better performance as separable functions; *Alpine function (*10*D)*, *Quintic global optimization problem*, *Stochastic global optimization problem (*10*D,* 20*D)*; and for multimodal functions *Alpine function (*10*D)*, *Quintic global optimization problem*, *Stochastic global optimization problem (*10*D,* 20*D)*. The CoDE algorithm has better performance for separable functions; *Rastrigin's function (*10*D,* 20*D)*, *Levy function (*10*D)*, *Alpine function (*20*D,* 30*D)*, *Neumaier*-2 *Problem (*30*D)*, *Deflected Corrugated Spring (*10*D,* 20*D)*, *MultiModal global optimization problem;* for non-separable function *Ackley's path function (*10*D,* 20*D)*, *Stretched-V global optimization problem*, *XinSheYang (*20*D,* 30*D);* for unimodal functions *Neumaier*-2 *Problem (*30*D)* and for multimodal functions; *Rastrigin's*

Table 1. Test Suit of Benchmark Functions (f_1– f_{16}).

Function	Name of Function (type)	Equation	Search Space	Optima
f_1	Sphere model (Separable, Multimodal)	$f(x) = \sum_{i=0}^{n} x_i^2$	$-5.12 \le x_i \le 5.12$	0
f_2	Axis parallel hyperellipsoid (Separable, Unimodal)	$f(x) = \sum_{i=0}^{n} i \cdot x_i^2$	$-5.12 \le x_i \le 5.12$	0
f_3	Schwefel's problem 1.2 (Non-Separable, Unimodal)	$f(x) = \sum_{i=0}^{n} \left(\sum_{j=0}^{i} x_j \right)^2$	$-65 \le x_j \le 65$	0
f_4	Rosenbrock's valley (Non-Separable, Unimodal)	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$-30 \le x_i \le 30$	0
f_5	Rastrigin's function (Separable, Multimodal)	$f(x) = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i))$	$-5.12 \le x_i \le 5.12$	0
f_6	Griewank's function (Non-Separable, Multimodal)	$f(x) = \sum_{i=1}^{n} (- \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) +$	$-600 \le x_i \le 600$	0
f_7	Sum of different power (Non-Separable, Multimodal)	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert^{(i+1)}$	$-1 \le x_i \le 1$	0
f_8	Ackley's path function (Non-Separable, Multimodal)	$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}} \right) - \exp\left(\frac{\sum_{i=i}^{n} Cos(2\pi x_i)}{n} \right) + 20 + e$	$-32 \le x_i \le 32$	0
f_9	Levy function (Separable, Multimodal)	$0.1\left[\sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 \times (1 + \sin^2(3\pi x_i + 1)) + (x_n - 1)(1 + \sin^2(2\pi x_n)) \right]$	$-10 \le x_i \le 10$	0
f_{10}	Zakharov function (Non-Separable, Multimodal)	$f(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=i}^{n} 0.5 i x_i \right)^2 + \left(\sum_{i=i}^{n} 0.5 i x_i \right)^4$	$-5 \le x_i \le 10$	0
f_{11}	Schwefel's problem 2.22 (Non-Separable, Unimodal)	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$-10 \le x_i \le 10$	0
f_{12}	Step function (Separable, Unimodal)	$f(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	$-100 \le x_i \le 100$	0
f_{13}	De Jong's function 4 (no noise) (Separable, Unimodal)	$f(x) = \sum_{i=1}^{n} i x_i^4$	$-1.28 \le x_i \le 1.28$	0
f_{14}	Alpine function (Separable, Multimodal)	$f(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1 x_i \rvert$	$-10 \le x_i \le 10$	0
f_{15}	Levy and Montalvo Problem (Separable, Multimodal)	$f(x) = \left(\frac{\pi}{n} \right) \left(10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin 2(\pi y_{i+1})] \right) + (y_n - 1)^2$ where	$-10 \le x_i \le 10$	0
f_{16}	Neumaier 2 Problem (Separable, Unimodal)	$f(x) = \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} (x_i x_{i-1})$	$-n^2 \le x_i \le n^2$	0

Table 2. Test Suit of Benchmark Functions (f_{17}—f_{30}).

Function	Name of Function (type)	Equation	Search Space	Optima
f_{17}	Cosine Mixture (Separable, Multimodal)	$f(x) = -0.1\sum_{i=1}^{n}\cos(5\pi x_i) + \sum_{i=1}^{n}x_i^2 \|\|x\|\| = \sqrt{\sum_{i=1}^{n}x_i^2}$	$-1 \le x_i \le 1$	-0.1x(n)
f_{18}	Cigar (Separable, Multimodal)	$f(x) = x_1^2 + 100,000\sum_{i=1}^{n}x_i^2$	$10 \le x_i \le 10$	0
f_{19}	Function "15" (Separable, Multimodal)	$f(x) = \sum_{i=1}^{n-1}\left[0.2x_i^2 + 0.1x_i^2\sin(2x_i)\right]$	$10 \le x_i \le 10$	0
f_{20}	Ellipse Function (Separable, Unimodal)	$f(x) = \sum_{i=1}^{n}(10^{6(\frac{i-1}{n-1})}x_i^2)$	$100 \le x_i \le 100$	0
f_{21}	Tablet Function (Separable, Unimodal)	$f(x) = 10^4x_1^2 + \sum_{i=2}^{n}x_i^2$	$100 \le x_i \le 100$	0
f_{22}	Schewel (Separable, Multimodal)	$f(x) = \sum_{i=1}^{n}((x_1 - x_i^2)^2 + (x_i - 1)^2)$	$32 \le x_i \le 32$	0
f_{23}	Deflected Corrugated Spring (Separable, Multimodal)	$f(x) = 0.1\sum_{i=1}^{n}\left((x_i - \alpha)^2 - \cos\left(K\sqrt{\sum_{i=1}^{n}((x_i - \alpha)^2)}\right)\right)$		0
f_{24}	Mishra 1 global optimization problem (Non-Separable, Multimodal)	$f(x) = (1 + x_n)^{x_n}$ where $x_n = n - \sum_{i=1}^{n-1}x_i$	$0 \le x_i \le 1$	2
f_{25}	Mishra 2 global optimization problem (Non-Separable, Multimodal)	$f(x) = (1 + x_n)^{x_n}$ where $x_n = n - \sum_{i=1}^{n-1}\frac{(x_i + x_{i+1})}{2}$	$0 \le x_i \le 1$	2
f_{26}	MultiModal global optimization problem (Separable, Multimodal)	$f(x) = \left(\sum_{i=1}^{n}\|x_i\|\right)\left(\prod_{i=1}^{n}\|x_i\|\right)$	$10 \le x_i \le 10$	0
f_{27}	Quintic global optimization problem (Separable, Multimodal)1	$f(x) = \sum_{i=1}^{n}\|x_i^5 - 3x_i^4 + 4x_i^3 + 2x_i^2 - 10x_i - 4\|$	$10 \le x_i \le 10$	-1
f_{28}	Stochastic global optimization problem (Separable, Multimodal)	$f(x) = \sum_{i=1}^{n}\epsilon_i\left\|x_i - \frac{1}{i}\right\|$	$5 \le x_i \le 5$	0
f_{29}	Stretched V global optimization problem (Non-Separable, Multimodal)	$f(x) = \sum_{i=1}^{n-1}t^{1/4}\left[\sin\left(50t^{0.1}\right) + 1\right]^2$ where $t = x_{i+1}^2 + x_i^2$	$10 \le x_i \le 10$	0
f_{30}	XinSheYang (Non-Separable, Multimodal)	$f(x) = \left(\sum_{i=1}^{n}\|x_i\|\right) \Big/ e^{\sum_{i=1}^{n}\sin(x_i^2)}$	$2\pi \le x_i \le 2\pi$	0

Table 3. 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**

Note: Bold values show the best values of algorithm as compared to other algorithms.

Table 4. Number of Function Call Results of Functions (f_{16}—f_{30}).

Function	DIM	DE	CoDE	EPSDE	jDE	RCPDE
f_{16}	10D	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
	20D	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
	30D	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**
f_{17}	10D	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**
	20D	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**
	30D	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**
f_{18}	10D	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**
	20D	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**
	30D	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**
f_{19}	10D	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**
	20D	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**
	30D	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**
f_{20}	10D	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**
	20D	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**
	30D	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**
f_{21}	10D	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**
	20D	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**
	30D	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**
f_{22}	10D	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**
	20D	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**
	30D	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
f_{23}	10D	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
	20D	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
	30D	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
f_{24}	10D	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
	20D	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**
	30D	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**
f_{25}	10D	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**
	20D	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**
	30D	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
f_{26}	10D	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
	20D	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
	30D	**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
f_{27}	10D	**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
	20D	**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
	30D	**3.74E+02±2.36E+01**	–	6.92E+02±4.35E+01	4.47E+02±3.23E+01	5.35E+02±3.55E+01
f_{28}	10D	**1.03E+03±4.33E+01**	–	1.99E+03±9.59E+01	1.09E+03±5.89E+01	1.26E+03±6.38E+01
	20D	1.98E+03±5.09E+01	–	3.73E+03±1.45E+02	**1.87E+03±1.05E+02**	2.13E+03±7.73E+01
	30D	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
f_{29}	10D	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
	20D	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
	30D	–	–	–	–	–
f_{30}	10D	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
	20D	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
	30D	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

Note: Bold values show the best values of algorithm as compared to other algorithms.

Table 5. 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**

Note: Bold values show the best values of algorithm as compared to other algorithms.

Table 6. 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**

Note: Bold values show the best values of algorithm as compared to other algorithms.

function (10*D*, 20*D*), *Ackley's path function* (10*D*, 20*D*), *Levy function* (10*D*), *Alpine function* (20*D*, 30*D*), *Deflected Corrugated Spring* (10*D*, 20*D*), *MultiModal global optimization problem*, *Stretched-V global optimization problem*, *XinSheYang* (20*D*, 30*D*). The jDE algorithm has better NFC performance for Separable functions; *Rastrigin's function* (30*D*), *Levy function* (20*D*, 30*D*), *Neumaier-2 Problem*, *Neumaier-2 Problem* (20*D*), *Deflected Corrugated Spring* (30*D*), *Stochastic global optimization problem* (30*D*); *for* Non-Separable functions; *Ackley's path function* (30*D*); for Unimodal functions; *Neumaier-2 Problem* (20*D*) and for Multimodal functions; *Rastrigin's function* (30*D*), *Ackley's path function* (30*D*), *Levy function* (20*D*, 30*D*), *Deflected Corrugated Spring* (30*D*), *and Stochastic global optimization problem* (30*D*).

Average fitness results of DE, EPSDE, CoDE, jDE and the proposed RCPDE are presented in Tables 5 & 6. From average fitness results, it can be observed that the fitness performance of the proposed RCPDE has better fitness performance in most cases. The proposed 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} (10*D*), 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} (10*D*), 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} (20*D*, 30*D*), f_{15}, f_{16} (10*D*), f_{17} (20*D*, 30*D*), f_{22} (20*D*), f_{26}, f_{27} (10*D*), f_{28} (10*D*, 20*D*); for non-separable functions; f_{24}, f_{25}; for unimodal functions; f_{12}, f_{13} (20*D*, 30*D*), f_{16} (10*D*) and multimodal functions; f_{15}, f_{17} (20*D*, 30*D*), f_{22} (20*D*), f_{24}, f_{25}, f_{26}, f_{27} (10*D*), f_{28} (10*D*, 20*D*). The jDE algorithm has better average fitness performance for separable functions; f_9, f_{12}, f_{13} (20*D*, 30*D*), f_{15}, f_{16} (10*D*), f_{17}, f_{23} (10*D*, 20*D*), f_{26}, f_{28} (30*D*); for non-separable functions; f_6 (20*D*), f_7 (20*D*, 30*D*), f_{24}, f_{25}, f_{29}, f_{30} (20*D*, 30*D*); for unimodal functions; f_{12}, f_{13} (20*D*, 30*D*), f_{16} (10*D*) and for multimodal functions; f_6 (20*D*), f_7 (20*D*, 30*D*), f_9, f_{15}, f_{17}, f_{23} (10*D*, 20*D*), f_{24}, f_{25}, f_{26}, f_{28} (30*D*), f_{29}, f_{30} (20*D*, 30*D*). The CoDE algorithm has better average fitness performance for separable functions; f_5, f_{12}, f_{13} (10*D*, 20*D*), f_{14}, f_{15}, f_{16}, f_{17}, f_{22}, f_{23} (30*D*), f_{26}, f_{27} (20*D*, 30*D*); for non-separable functions; f_4 (10*D*), f_6, f_7, f_8, f_{24}, f_{25}, f_{30} (10*D*, 20*D*); for unimodal functions; f_4 (10*D*), f_{12}, f_{13} (10*D*, 20*D*), f_{16} and for multimodal functions; f_5, f_6, f_7, f_8, f_{14}, f_{15}, f_{17}, f_{22}, f_{23} (30*D*), f_{24}, f_{25}, f_{26}, f_{27} (20*D*, 30*D*), f_{30} (10*D*, 20*D*). The EPSDE algorithm has better average fitness performance for separable functions; f_5, f_{12}, f_{13}, f_{15}, f_{16} (10*D*), f_{17}, f_{22}, f_{26}; for non-separable functions; f_4, f_6, f_7 (20*D*, 30*D*), f_{24}, f_{25}, f_{30} (10*D*, 20*D*); for unimodal functions; f_4, f_{12}, f_{13}, f_{16} (10*D*) and for multimodal functions; f_5, f_6, f_7 (20*D*, 30*D*), f_{15}, f_{17}, f_{22}, f_{24}, f_{25}, f_{26}, f_{30} (10*D*, 20*D*). The overall results of NFC and average fitness indicates that the performance of proposed RCPDE is dominating for unimodal, multimodal separable and non-separable functions.

Figure 1 contains logarithmic convergence graphs of selected functions showing iterations horizontally and performance vertically. Convergence graphs of DE, EPSDE, CoDE, jDE, the proposed RCPDE generated for 20D average fitness values. The convergence graphs of f_1, f_7, f_{13} and f_{18} in Figure 1 contains an average fitness convergence graph of the proposed RCPDE and other state of the art DE algorithms for *Sphere model* (f_1), *Sum of different power* (f_7), *De Jong's function* 4 (f_{13}) and *Cigar* (f_{18}) functions. The convergence graph of f_1, f_7, f_{13} and f_{18} depicts that the proposed RCPDE has quick convergence from starting iteration till final iteration among all other algorithms. In all these cases the proposed RCPDE reaches at optimal value 0 within the given iterations more quickly than the other algorithms. The convergence graphs of f_3 and f_{10} for *Schwefel's problem* 1.2 (f_3) and *Zakharov function* (f_{10}) also shows quick convergence of the proposed RCPDE algorithm than all other algorithms throughout the execution cycle. So it is clear from convergence graphs that the performance of proposed RCPDE is better than DE, EPSDE, CoDE and jDE. The proposed RCPDE will prove to be significant addition in DE literature.

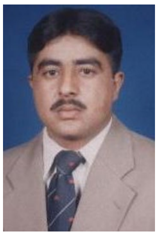
Conclusion

This research work proposes a novel random controlled, pool based selection differential evolution (RCPDE) algorithm. The proposed mutation strategy and control parameter pools are found to be highly beneficial in balancing the exploration and exploitation ability of the DE algorithms by incorporating potential mutation strategies and diverse control parameter values in the DE algorithm. Two commonly used performance metrics NFC and Average fitness values are used to compare the performance of the proposed RCPDE with other state of the art DE algorithms. The proposed algorithm is tested through simulation and results are reported in various parts of this paper. Simulation results have shown that the proposed RCPDE improves the solution quality as well as convergence speed of DE algorithms. This research work can be further enhanced by incorporating the memorization of convergence track over time with both parameter pool and mutation strategy pool.

Conflict of interest statement

The authors declare that there is no conflict of interests regarding the publication of this paper.

Notes on Contributors



Qamar Abbas received his BSc in Mathematics and Statistics in 2002 from B.Z.U Multan, Pakistan and received an MSc in Computer Sciences in 2004 from Gomal University D.I.Khan. He also received an MS Degree in Computer Science in 2008 and PhD in computational intelligence in 2016 from Iqra University Islamabad, Pakistan. Currently, he is working in the Department of Computer Science and Software Engineering as a faculty member at the International Islamic University Islamabad. His research interests have been centered around learning algorithms, evolutionary algorithms, data mining and artificial neural networks.

Jamil Ahmad received his BSc in Mathematics and Physics in 1985 from Islamia College Peshawar, Pakistan and received an MSc in Computer Sciences in 1989 from University of Peshawar. He also received an MSc in Information Technology in 1992 from University of Warwick, UK and a PhD in Artificial Neural Networks in 1995 from King's College London, UK. In 1995 he joined GIK Institute, Pakistan where he was also appointed as Dean of Faculty of Computer Engineering. Currently, he is working as Professor/

Vice Chancellor of Abasyn Universities Peshawar and Islamabad campuses. Since his first publication in 1991, his research interests have been centered around learning algorithms especially for pattern recognition and hand written character recognition. He is a senior member of IEEE and fellow member of the British Computer Society. He has also awarded a CEng status.

Hajira Jabeen (F) is a senior researcher at the University of Bonn. Her research interests include Machine Learning, Semantic Web, Data Mining, Evolutionary Computation and Big Data. She has broad international experience of teaching, administration and research. She has worked on many Machine Learning and Data Mining projects, including working as the work package lead in H2020 funded project, Big Data Europe. She has authored several articles in prominent international journals & conferences, and also served as a PC member in a number of conferences.

References

Abbas, Q., Ahmad, J., & Jabeen, H. (2015). A novel tournament selection based differential evolution variant for continuous optimization problems. *Mathematical Problems in Engineering*, 2015, 1–21. Article ID 205709, doi:10.1155/2015/205709

Ali, M., Pant, M., & Abraham, A. (2009). Simplex differential evolution. *Acta Polytechnica Hungarica, 6*, 95–115.

Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation, 10*, 646–657.

Brown, C., Jin, Y., Leach, M., & Hodgson, M. (2016). μ JADE: adaptive differential evolution with a small population. *Soft Computing, 20*(10), 4111–4120.

Buhry, L., Giremus, A., Grivel, E., Saïghi, S., & Renaud, S. (2009). *New variants of the differential evolution algorithm: Application for neuroscientists*. 17th European Signal Processing Conference (pp. 2352–2356). Glasgow, Scotland.

Chowdhury, A., Giri, R., Ghosh, A., Das, S., Abraham, A., & Snasel, V. (2010). *Linear antenna array synthesis using fitness-adaptive differential evolution algorithm*. IEEE Congress on Evolutionary Computation (pp. 18–23). Bercelona, Spain.

Das, S., Abraham, A., Chakraborty, U.K., & Konar, A. (2009). Differential evolution using a neighborhood-based mutation operator. *IEEE Transactions on Evolutionary Computation, 13*, 526–553.

Das, S., Abraham, A., & Konar, A. (2008). Particle swarm optimization and differential evolution algorithms: Technical analysis, applications and hybridization perspectives. *Studies in Computational Intelligence, 11*, 1–38.

Dong, X.L., Liu, S.Q., Tao, T., Li, S.P., & Xin, K.L. (2012). A comparative study of differential evolution and genetic algorithms for optimizing the design of water distribution systems. *Journal of Zhejiang University Science A, 13*, 674–686.

Dragoi, E.N., Curteanu, S., Galaction, A.I., & Cascaval, D. (2013). Optimization methodology based on neural networks and self-adaptive differential evolution algorithm applied to an aerobic fermentation process. *Applied Soft Computing, 13*, 222–238.

Engelbrecht, A.P. (2007). *Computational intelligence an introduction* (2nd Edition ed.). Chichester, UK: Wiley.

Godfrey, C.O., & Donald, D. (2009). *Differential evolution: A handbook for global permutation-based combinatorial optimization.* (J. Kacprzyk, Ed.). Heidelberg: Springer.

Gong, W., Cai, Z., Ling, C.X., & Li, H. (2011). Enhanced differential evolution with adaptive strategies for numerical optimization. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 41*, 397–413.

Guo, S.M., Yang, C.C., Hsu, P.H., & Tsai, J.S. (2015). Improving differential evolution with a successful-parent-selecting framework. *Evolutionary Computation, 19*, 717–730.

Hao, X., Chen, D., Wu, X., & Yu, H. (2004). Generalized regression neural network and its application to delayed coking process. *Journal of Chemical Industry and Engineering(China), 55*, 608–612.

Huang, L., He, D., & Yang, S.X. (2013). Segmentation on ripe Fuji apple with fuzzy 2D entropy based on 2D histogram and GA optimization. *Intelligent Automation & Soft Computing, 19*, 239–251.

Iorio, A. W., & Li, X. (2004, December). Solving rotated multi-objective optimization problems using differential evolution. In Australasian Joint Conference on Artificial Intelligence (pp. 861–872). Heidelberg: Springer.

Islam, SM., Das, S., Ghosh, S., Roy, S., & Suganthan, P.N. (2012). 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, 42*, 482–500.

Kacprzyk, J. (2008). *Studies in computational intelligence, Volume 155.* (H.R. Tizhoosh, & M. Ventresca, Eds.). New York, NY: Springer.

Li, X., & Yin, M. (2016). Modified differential evolution with self-adaptive parameters method. *Journal of Combinatorial Optimization, 31*, 546–576.

Liu, J., & Lampinen, J. (2005). A fuzzy adaptive differential evolution algorithm. *Soft Computing, 9*, 448–462.

Liu, G., Li, Y., & He, G. (2010). *Design of digital FIR filters using differential evolution algorithm based on reserved genes*. IEEE Congress on Evolutionary Computation (CEC) (pp. 1–7). Barcelona.

Mallipeddi, R., & Suganthan, P.N. (2010). *Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies*. First International Conference on Swarm, Evolutionary, and Memetic Computing, SEMCCO 2010, December 16–18. (pp. 71–78). Chennai, India.

Mallipeddi, R., Suganthan, P.N., Pan, Q.K., & Tasgetiren, M.F. (2011). Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing, 11*, 1679–1696.

Marchiori, E., Moore, J.H., & Rajapakse, J.C. (2007). *Evaluating evolutionary algorithms and differential evolution for the online optimization of fermentation processes*. EvoBIO (pp. 236–246). Valencia, Spain.

Maulik, U., & Saha, I. (2009). September). Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery. *Pattern Recognition, 42*, 2135–2149.

Mezura-Montes, E., Reyes, J.V., & Coello Coello, C.A. (2006). A comparative study of differential evolution variants for global optimization. *Genetic and Evolutionary Computation Conference (GECCO)* (pp. 485–492). Washington, USA.

Nama, S., Saha, A.K., & Ghosh, S. (2016). A new ensemble algorithm of differential evolution and backtracking search optimization. *Industrial Engineering Computations, 7*, 323–338.

de Oliveira, G. T. S. & Saramago, S. F. P. (2008). A contribution to the study about differential evolution. *Ciência & Engenharia, 16*(1/2), 01–08.

Price, K., Storn, R.M., & Lampinen, J.A. (2005). *Differential evolution: A practical approach to global optimization (natural computing series)* (1st Edition ed.). New York, NY: Springer-Verlag.

Qin, AK., Huang, V.L., & Suganthan, P.N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation, 2*, 398–417.

Rahnamayan, S., Tizhoosh, H.R., & Salama, M.M. (2008). Opposition-based differential evolution. *IEEE Transactions on Evolutionary Computation, 12*, 64–79.

Ronkkonen, J., Kukkonen, S., & Price, K.V. (2005). Real-parameter optimization with differential. *IEEE Congress on Evolutionary Computation* (pp. 506–513). Edinburgh, Scotland, UK.

Segura, C., Coello, C.A., Segredo, E., & Leon, C. (2015). On the adaptation of the mutation scale factor in differential evolution. *Optimization Letters, 9*(1), 189–198.

Smirnov, A., & Jastrzebski, R.P. (2009). *Differential evolution approach for tuning an H∞controller in AMB systems*. 35th IEEE Annual Conference of Industrial Electronics (pp. 1514–1518). Lappeenranta, Finland.

Storn, R. & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization, 11*(4), 341–359.

Storn, R., & Price, K. (1995). *Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces.* Rep, CA, Berkeley: Tech.

Velagic, J. & Osmic, J. (2013). Fuzzy-genetic identification and control stuctures for nonlinear helicopter model. *Intelligent Automation & Soft Computing, 19*, 51–68.

Wang, Y., Cai, Z., & Zhang, Q. (2011). Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation, 15*, 55–66.

Xu, X., & Li, Y. (2007). Comparison between particle swarm optimization, differential evolution and multi-parents crossover. IEEE International Conference on Computational Intelligence and Security, (pp. 124–127). Harbin, China.

Yao, X., Liu, Y., & Lin, G. (1999, July). Evolutionary programming made faster. *IEEE Transaction on Evolutionary Computation, 3*, 82–102.

Yuan, X., Wang, L., Zhang, Y., & Yuan, Y. (2009). A hybrid differential evolution method for dynamic economic dispatch with valve-point effects. *Expert Systems with Applications, 36*, 4042–4048.

Zaharie, D. (2003). Control of population diversity and adaptation in differential evolution algorithms. *In proceeding of MENDEL, 9*, 41–46.

Zhang, J., & Sanderson, A.C. (2009). JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation, 13*, 945–958.