Mathematical Modelling of Engineering Problems
Vol. 9, No. 4, August, 2022, pp. 1017-1024
Journal homepage: http://iieta.org/journals/mmep

An Overview of Self-Adaptive Differential Evolution Algorithms with Mutation Strategy

Dania AlKhulaifi, Maryam AlQahtani, Zenab AlSadeq, Atta-ur Rahman*, Dhiaa Musleh

Department of Computer Science (CS), College of Computer Science and Information Technology (CCSIT), Imam Abdulrahman Bin Faisal University (IAU), P.O. Box 1982, Dammam 31441, Saudi Arabia

Corresponding Author Email: aaurrahman@iau.edu.sa

https://doi.org/10.18280/mmep.090419

Received: 24 Janurary 2022
Accepted: 2 August 2022

Keywords:
differential evolution, self-adaptive differential evolution, mutation strategy, optimization

ABSTRACT

Differential Evolution (DE) is a widely used global searching algorithm that solves real-world optimization problems. It is categorized as a stochastic parameter optimization method that has a broad spectrum of applications, notably neural networks, logistics, scheduling, and modeling. In practice, different optimization issues need different parameter settings. Due to DE simplicity, ease of implementation, and dependability, many scientists were interested in examining this algorithm. Nonetheless, the quality of DE and its variations are directly influenced by different mutation techniques and control parameter settings. In this paper, an overview and analogy of some algorithms that employ different mutation techniques will be illustrated. Additionally, a novel strategy that uses different mutation methods is proposed and compared with some existing strategies.

1. INTRODUCTION

The existence of Global Optimization Problems (GOP) motivated researchers to develop methods that employ mathematical optimization and probabilistic algorithms. Most mathematical optimization programs feed on gradient functions, as non-differentiable problems require different methods. Furthermore, when handling complex optimization problems, mathematical programming approaches are vulnerable to being locked in a local optimum which keeps possible better solutions undiscovered [1]. Several real-world problems such as nonlinear, multimodal, hybrid, composite, and large-scale problems are complex by nature [2]. Experimental results showed that the meta-heuristic method performs well in tackling such problems, attracting scholars to conduct a further investigation [3]. Most meta-heuristic algorithms are based on natural phenomena. On the contrary, typical mathematical programming methods replicate natural phenomena using established rules and randomization methods to avoid gradient difficulties in optimization. Many studies on metaheuristic approaches have been done over the last two decades [4, 5]. Additionally, multiple metaheuristic algorithms have been developed to address continuous problems, including the Genetic Algorithm [6, 7], Differential Evolution [8-10], and Artificial Bee Colony [11]. Nonetheless, DE has its own flagship and superiority over the peer approaches due to its applicability in both discrete and continuous spaces, less vulnerability to stuck in the local minima and verstaile nature of adaptability in terms of stochastic operator selection and fine tuning in fusion with other secondary algorithms, a case of memetic computing [1]. The use of "trial-and-error" to obtain optimal answers is the main feature of these algorithms. As a result, these algorithms have shown success in addressing global optimization issues [12]. Storn & Price in 1995 [13] established the DE strategy, which is a basic yet effective population-based non-linear search method that works as a global optimizer in a continuous search area. Engineering, communication [8-10], and pattern recognition are just a few of the fields where DE has been applied successfully. There are various advantages of DE over the other evolutionary algorithms of similar nature like GA, PSO, ACO etc. Such as, it is equally applicable and successful in discrete and continuous spaces problems. It has less vunerability to trap in local minima as GAs are more vulnerable to it. It allows fine tuning and gaurantees the convergance with fast tapper off rate. Being a global optimizer, it also have capabilities of being hybridized with local optimizers like pattern search (PS). Moreover, it is adaptive and flxible in terms of operator selection. All these features makes it prominent candidate over its other counterparts. There are many trial vector generation techniques in DE, but only a few of them can be acceptable for solving a specific problem where variables might vary. The DE algorithm's performance is usually determined by the mutation method, crossover method, and control parameters. In several optimization problems, applying the same function with adjusting criteria of time consumption and accuracy yields different results; hence, ideal values for the control parameters might vary. As a result, solving a given optimization problem successfully requires a long trial-and-error search to find an optimal consolidation of techniques and parameter values. Some improvements based on control parameter adaptation strategy were proposed to address this issue, including fuzzy adaptive DE [14], self-adaptive chaos DE [15], an adaptive DE (ADE) [16], a self-adaptive DE [17], and adaptive DE with modified parameters [18]. To adjust this algorithm there are multiple mutation techniques that provide different features. Moreover, (DE/rand/1, DE/best/1, and DE/current-to-best/1) techniques have a higher capacity for local exploitation and a faster convergence rate [19]. As a result, they are more suited for handling unimodal situations. When solving multimodal problems, (DE/rand/1 and DE/rand/2) have a better global search capability [20]. Some DE variants focus on incorporating different mutation methods [21], such as a self-adaptive mutation DE [22], a composite DE [23], an ensemble of mutation strategies and control parameters of DE [24], and a self-adaptive DE with discrete mutation control parameters. This paper aims at discussing some existing enhancing algorithms and proposing a new modification. Section I, illustrates the classical scheme working mechanism. Followed by section II, which discusses previous work and reviewed literature. In addition, a comparison of the DE adopting algorithms is given in section III. Section IV describes the modified algorithm. In the last section V, the findings of this article will be summarized and concluded.

1.1 Classical differential evolution algorithm

DE is a basic nature-inspired algorithm that solves optimization problems. One of its main features is the ease of implementation and study. It operates in four basic steps [25] as depicted in Figure 1.

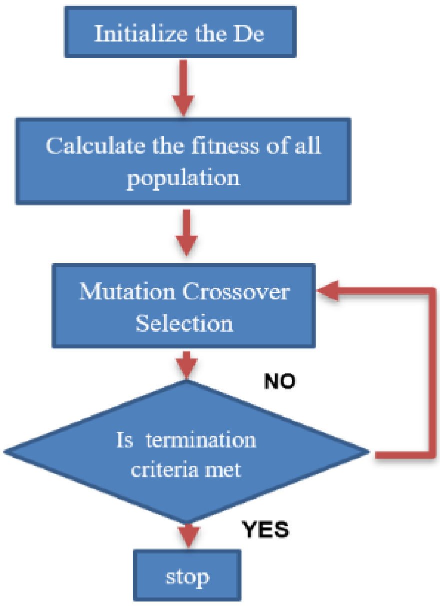

Figure 1. Working scheme of classical DE

Step 1: Initialization

DE will find an optimal solution in a multidimensional continuous space by randomly initializing the population of the given problem. Those parameter vectors are thought to be like chromosomes that produce the final solution i.e. Scale Factor (F), Crossover Rate (CR), Population size (NP), and a number of iterations (I).

Step 2: Mutation

In this step, a trial vector v_i^{w+1} is produced by the evolution of a target vector. Typically, v_i^{w+1} is found by one of the below equations:

$$v_i^{W+1} = X_{i3}^K + f(x_{i1}^w - x_{i2}^w) \tag{1}$$

$$v_i^{W+1} = X_{best}^K + f(x_{i1}^w - x_{i2}^w) \tag{2}$$

$$v_i^{W+1} = X_i^K + F1(x_{best}^w - x_i^w) + F2(x_{i1}^w - x_{i2}^w) \tag{3}$$

$$v_i^{W+1} = X_{best}^K + F1(x_{i1}^w - x_{i2}^w) + F2(x_{i3}^w - x_{i4}^w) \tag{4}$$

$$v_i^{W+1} = X_{i5}^K + F1(x_{i1}^w - x_{i2}^w) + F2(x_{i3}^w - x_{i4}^w) \tag{5}$$

where, *F*, *F1* and *F2* are scale factors x_{beas}^w indicates the ultimate vector in iteration (wi(n)) and those represent different integer values that are randomly selected from the set {1,2…, N}.

Step 3: Crossover

After generating a donor vector, a crossover step takes place. This step applies exponential and binomial crossover mechanisms. The method of crossover is as follows:

$$u_{ij}^{w+1} = \begin{cases} v_{ij}^{w+1} \; if\ rand() < CR\ or\ j = r \\ v_{ij}^{w} \qquad\qquad else \end{cases} \tag{6}$$

Step 4: Selection

The final step is to decide if the target and trial vectors would pass to the next generation. When the stopping condition is reached as in the maximum number of iterations, the algorithm terminates, or else it will go back and repeat from step 3. The selection method operates as follows:

$$\begin{gathered} \overrightarrow{x_{\iota,G+1}} = \vec{U}_{i,L} \\ iF\ f(\overrightarrow{U_{\iota,L}}) \le f(\overrightarrow{X_{\iota,L}}) = \vec{X}_{i,L} \\ iF\ f(\overrightarrow{U_{\iota,L}}) > f(\overrightarrow{X_{\iota,L}}) \end{gathered} \tag{7}$$

2. LITERATURE REVIEW

The section provides a brief but comprehensive overview of plain DE algorithm as well as it most widely used and promising (in terms of applications) variants.

Fan and Yan [26] proposed a DE-based solution dedicated for constrained optimization problems. The method enables each parent to produce several offspring by applying a different mutation operator. The proposed strategy increases the probability of parents producing better solutions. Results showed an acceptable average computational cost and superior performance with some best previously developed solutions.

DE/rand/1/bin is the standard DE algorithm proposed by [13]. After that, Qin et al. [17] presented (SaDE) which uses a novel mutation mode (DE/current-to-pbest/1). Two new parameters have been added. The c parameter determines the rate of adaptation, while the p parameter determines the greediness of the mutation strategy. In SaDE, trial vector generation procedures, as well as their two control factors, have been probabilistically assigned to each target vector in the present population, depending on the probabilities gradually learned from experiments to produce better solutions. Experiments revealed that the SaDE algorithm could evolve appropriate methods and parameter values as evolution progressed. Also, the reduced root-mean-square deviation was attained by this variant. There are two new parameters added.

The rate of adaptation is controlled by the c parameter, whereas the greediness of the mutation strategy is determined by the p parameter. Wu et al. [23] proposed CoDE which uses three trial vector generating procedures which are: (i) “rand/1/bin”; (ii) “rand/2/bin”; (iii) “current-to-rand/1” and sets three control parameter which are: (1) F=1.0 and CR=0.1; (2) F=1.0 and CR=0.9; (3) F=0.8 and CR=0.2. CODE creates trial vectors by combining them at random. All the CEC2005 contest test instances have been tested with CoDE. Results yielded competitive results. Ronkkonen et al. [22] proposed SAMDE, which uses an adaptive mutation operator to provide the benefits of the DE/rand/1/bin method and the DE/best/2/bin approach. This method was made to solve a design problem as a constrained minimization problem. As a result, its convergence performance was significantly enhanced. The findings are supported by numerical experiment data. Parameters in EPSDE which is a DE consisting of an ensemble of mutation and crossover techniques, as well as their control factors. Throughout the evolution process, combinations of different mutation and crossover techniques, in addition to a group of values for each control parameter, compete to create offspring in EPSDE. An attempt to tune Scale Factor (F) and Crossover Rate (Cr) depending on the objective function value in the population level without user input. This method yielded success in accuracy, robustness, convergence, and speed compared with other previously developed variants. Moreover, the ensemble of mutation strategies and Parameters in DE [27] were employed to solve numerical optimization problems. In this variation, different distinct mutation strategies coupled with values for each control parameter are adjusted throughout the evolution process. This set competes to produce offspring. The performance of EPSDE was found to be better on a set of bound-constrained problems and compared with conventional DE and different variations of DE. Zou et al. [28] aim at solving unconstrained optimization problems by proposing a modified version of the DE algorithm (MDE). To tune crossover rates and scale factors, their method increases the diversity of possible solutions by the guess distribution and uniform distribution. Also, an external archive with dynamic probabilistic methods was used to ensure quality. Additionally, new solutions are created during the late evolution stage which improves the convergence of the algorithm. After considering other candidate solutions, a middle solution will be generated. It was proven that the MDE algorithm gets better objective function values, hence it can be employed to solve unconstrained optimization problems. Wu et al. [29] presented a multi-population-based approach that ensembles multiple strategies, resulting in the Multi-Population Ensemble DE (MPEDE), in which the population is divided into three subpopulations of similar size and one larger subpopulation The three populations are exposed to three different mutation strategies: "current-to-pbest/1," "current-to-rand/1," and "rand/1," with the controlling parameters of each method made adaptively. A self-adaptive DE method with discrete mutation control parameters was proposed by Fan and Yan [26] (DMPSADE). This ensures that each variable has its mutation control parameter, as well as a crossover control parameter and mutation technique for each individual. For handling unconstrained global real parameter optimization in the continuous domain, Li and Yin [30] suggested a modified (DE) with self-adaptive parameter sets. This method employs two mutation criteria based on the population's rand and best individuals. Mutation, crossover, and greedy selection are three types of operators used in this approach to identify solutions. The parameter selection and mutation strategy are significant in resolving an issue. As a result, this algorithm can be superior to others. Mallipeddi et al. [31] proposed (MPMSDE) which instead of using the (MPEDE) grouping method, will create a new grouping approach that uses the ranking of strategies to allocate resources to the most suitable strategy. In addition, to avoid dropping into a local optimum, an information-sharing mechanism will be used in the largest sub-population. Additionally, the "DE/pbad-to-pbest-to-gbest/1" mutation will be used to replace the MPEDE method. To update individuals, the proposed method shows the global best solution which is a strong factor compared to MPEDE. The newly developed technique has proven to balance exploration and exploitation while yielding increased convergence. The results of the experiments reveal that the MPMSDE algorithm's performance is competitive on specific functions. The proposed MPMSDE method has outstanding performance in solving some issues, hence it is important.

Wang et al. [32] proposed a Self-adaptive DE algorithm with enhanced mutation mode (IMMSADE). Each individual has their control settings which are dynamically and rigourously changed/updated based on the population diversity and the individual differences among the produced offsprings. Sun et al. [33] proposed a novel simple variant of the DE Time Varying (TVDE) algorithm. It is used in numerical optimization problems and can be applied for multimodal optimization, and dynamic optimization. In TVDE**,** Three functions with time-varying features are used to develop a new mutation operator. Also, two parameters are automatically set during the process**.** For quality assurance, the proposed TVDE was evaluated against seven DE versions. The result demonstrates that among the eight DE algorithms, the TVDE method achieves the best overall performance. Sun et al. [34] proposed a new DE version (CSDE). Two mutation operators with altered features were used to generate the mutant vector. A coordination mechanism based on the past success rate was employed to adjust the two mutation operators. Results show that CSDE was better than seven other DE variants in most cases. Ali et al. [35] proposed a new technique that enables differential evolution algorithms to solve binary-based problems, such as binary knapsack. The method includes a mapping mechanism, representation of possible solutions, and diversity technique. Additionally, the fitness evaluation method was enhanced along with improved means of nominating candidate solutions. The algorithm was tested on 44 binary knapsack instances, and it has shown success in terms of computational time and finding better solutions for bigger knapsack problems. There are plenty of applications of DE in different areas of research [36-46].

The research of multi-objective DE algorithm and hybrid algorithm is a relatively new topic in DE. That is why it is among the most research intensive area to be explored in terms of its application in the multi-objective problems domain of hybrid [47-50] and fused intelligent systems such as proposed by Mahmud et al. [51] respectively. In multi-objective domain, the solution search spaces become huge due to the lenght of objective vector’s dimension [52-60]. For example, in case of binary/discrete problem, its complexity become exponential $O(2^n)$ and in case of continuous domain its complexity is even worst that is super-exponential $O(n^n)$. This actually makes the room for DE like algorithms to be investigated with all their poweful varaints. Table 1 presents the comparison of different DE based algorithm.

Table 1. Comparison of various DE algorithms

Ref.	Proposed Algorithm	Mutation method	Results
[13]	DE	DE/rand/1/bin	It was assumed to be the finest at the time, but it would have conveyed an excellent role.
[25]	Modified DE for constrained optimization	DE/rand/1/bin	Improved result quality, reduced average computational cost.
[17]	SADE	"DE/rand-to-best/1/bin," "DE/best/1/bin," and "DE/best/2/ bin"	It was more effective in obtaining higher success rates and higher quality solutions, with better stability and reduced value of standard deviation.
[23]	CoDE	Composite three vector generation schemes and three control parameters	The analysis indicated that it performed better than the other competitors in terms of total performance.
[22]	SAMDE	E/rand/1/bin strategy and the DE/best/2/bin	The findings of the computational studies revealed that the SAMDE methodology outperforms earlier design methodologies.
[24]	EPSDE	(DE/rand/1/bin)	The effectiveness of the proposed technique was favorable when compared to classifical DE methods
[28]	MDE	DE/rand/1/bin DE/best/1/bin	Convergence rate is high, and exploitation capacity is increased.
[26]	DMPSADE	It dynamically modifies mutation technigues and control parameters by competition.	The statistical findings reveal that DMPSADE's average performance outperforms all other competitors.
[29]	MPEDE	It made up of three different mutation strategies, "current-to-pbest/1" and "current-to-rand/1" and "rand/1".	It enhanced the adaptation of DE based on multiple population.
[30]	MDE	"DE/rand/1/bin".	The algorithm performs much better than the approaches in the literature, or is at least similar to it.
[31]	IMMSADE	Improving "DE/rand/1" mutation mode of the basic DE.	In terms of overall performance, IMMSADE exceeds the base DE and other DE algorithms.
[32]	TVDE	DE/rand/1, DE/best/1, DE/current-to-best/1, DE/best/2, and DE/rand/2.	Among the eight DE algorithms, the TVDE algorithm has the best overall performance.
[33]	CSDE	DE/current-to-pbest/1.	In most cases, CSDE outperforms seven state-of-the-art DE versions.
[35]	NBin-DE	DE/rand/1	Better average profit value, less computational time up to 94.4% improvement.
[12]	MPMSDE	Improving of MPEDE, DE/pbad-to-pbest-to-gbest/1	The MPMSDE algorithm performs much better than competition in terms of calculation accuracy and convergence rate, according to the results.

3. PROPOSED TECHNIQUE

During different phases of evolution, several mutation strategies with several parameter settings might give better results compared with using a single mutation strategy, as in the typical DE. Based on these insights, the proposed technique is made up of a pool of mutation and crossover techniques, as well as a pool of values for each of the control parameters. Every member of the starting population is given a mutation strategy and associated parameter values from one of the pools at random. Members of the population (the Target Vectors) create children (the Trial Vectors) using the mutation technique with parameter values provided to them. If the created trial vector outperforms the target vector, the proposed mutation technique and consequent parameters' values are carried over to the following generation's parents (Target Vector) and so on. The proposed mutation technique and parameter adjustment that created a better child vector in contrast to the parent vectors is saved. In case the target vector performs better than the trial vector, it is reinitialized at random with a the proposed mutation technique and consequently the related parameters' values from the corresponding pools or the winning combinations are saved with an equal/flat probability. This technique results in a higher possibility of producing offsprings (children/output vectors) in the future generations due to a greater mixture of various mutation approaches and/or operators and the related control parameters with more diversity and variations.

3.1 Optimization strategy

In this paper, the proposed method uses JADE and DE/pbad-to-pbest-to-gbest/1 mutation strategy to grant improved convergence and make use of globally optimal solutions. Binomial and exponential crossover algorithms with 100 iterations are utilized as crossover strategies, (-5.0,5.0) lower and upper bounds, a constant population size of (50) while mutation factor was $\in\{0.5\}$ and a crossover rate of $\in\{0.7\}$. The main purpose of this study is to investigate the said combination of the mutation operators and crossover techniques in the traditional DE algorithm to study its convergance rate and the strength to achieve the global optimum. Consequently, the said combination can be used for the problems with multi-objective cost functions optimization with more effectiveness compared to other variants and counterparts.

3.2 Implementation

This method used an objective function which is obj(x). All steps were combined in a differential evolutionary process. The population size, the boundaries of each input variable, the total number of iterations, the mutation scale factor, and the crossover rate are all input variables to this function. The function will return the most suitable option discovered and its evaluation. The coding implementation of the proposed variant of the differential evolution algorithm was conducted in Python language.

4. RESULTS AND DISCUSSION

After multiple trials, the results were drawn in a graph depicted in Figure 2. It reveals the objective function evaluation after each improvement. That is the algorithm exhibits with large changes initially (transient mode) and very small changes towards the end of the search (steady state) as the algorithm converged to the optima. The two dimensional plot shows the objective function evaluation for each improvement, as the algorithm converged on the optima. There were big changes at first and then very modest changes towards the end of the search. The trend is natural and realistic in terms of its convergence rate and no anomaly can be observed. In the early iterations the error is more nonetheless as the number of iterations increase from 5, the error reduces to zero for the standard objective function obj(x) where x is the trial vector being generated by the proposed methodology.

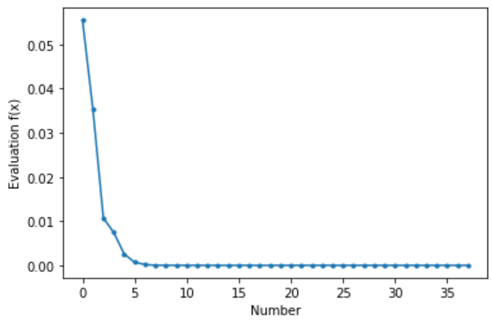

Figure 2. Performance convergence analysis

As shown in Figure 3, the objective function values are returned at every iteration which can be depicted by modifying the *main differential evolution* function to keep track of the values of the objective function or the cost function and return them in the list, namely, objective iterative. Finally the solution is found and the error goes to zero ultimately that shows no more need to evolute the algorithm. In Table 2, the proposed approach has been compared with two standard variants namely EPSDE and MPMSDE, in terms of standard deviation and mean. That are the two figure-of-merits to justify the performance of such type of stochastic algorithms in the literature. Consequently, it is evident that the proposed algorithm outperforms both EPSDE and MPMSDE in terms of both metrics. The comparison is fair in a way that same objective function has been used under same simulation environment and conditions.

```
Iteration: 1 f([[-0.2278  -0.06017]]) = 0.05551
Iteration: 2 f([[-0.18418  0.03751]]) = 0.03533
Iteration: 3 f([[-0.07416  0.07216]]) = 0.01071
Iteration: 6 f([[ 0.06192 -0.06017]]) = 0.00745
Iteration: 7 f([[-0.01954  0.04664]]) = 0.00256
Iteration: 9 f([[-0.00695  0.02555]]) = 0.00070
Iteration: 10 f([[-0.01304  0.00381]]) = 0.00018
Iteration: 14 f([[0.00248 0.00381]]) = 0.00002
Iteration: 17 f([[0.002   0.00323]]) = 0.00001
Iteration: 18 f([[0.00143 0.0008 ]]) = 0.00000
Iteration: 20 f([[-0.00046  0.00136]]) = 0.00000
Iteration: 22 f([[-0.00101 -0.00046]]) = 0.00000
Iteration: 23 f([[ 8.0e-05 -3.8e-04]]) = 0.00000
Iteration: 24 f([[-2.0e-05  2.3e-04]]) = 0.00000
Iteration: 25 f([[-2.e-04 -1.e-05]]) = 0.00000
Iteration: 26 f([[-4.e-05 -5.e-05]]) = 0.00000
Iteration: 28 f([[3.e-05 2.e-05]]) = 0.00000
Iteration: 29 f([[ 0.e+00 -1.e-05]]) = 0.00000
Iteration: 31 f([[-0.e+00  1.e-05]]) = 0.00000
Iteration: 38 f([[ 0. -0.]]) = 0.00000
Iteration: 40 f([[-0.  0.]]) = 0.00000
Iteration: 43 f([[-0. -0.]]) = 0.00000
Iteration: 49 f([[-0.  0.]]) = 0.00000
Iteration: 55 f([[-0. -0.]]) = 0.00000
Iteration: 59 f([[-0. -0.]]) = 0.00000
Iteration: 64 f([[-0.  0.]]) = 0.00000
Iteration: 66 f([[ 0. -0.]]) = 0.00000
Iteration: 68 f([[-0.  0.]]) = 0.00000
Iteration: 72 f([[ 0. -0.]]) = 0.00000
Iteration: 74 f([[-0.  0.]]) = 0.00000
Iteration: 76 f([[ 0. -0.]]) = 0.00000
Iteration: 80 f([[-0. -0.]]) = 0.00000
Iteration: 84 f([[-0.  0.]]) = 0.00000
Iteration: 88 f([[0. 0.]]) = 0.00000
Iteration: 91 f([[-0. -0.]]) = 0.00000
Iteration: 93 f([[ 0. -0.]]) = 0.00000
Iteration: 94 f([[0. 0.]]) = 0.00000
Iteration: 99 f([[-0.  0.]]) = 0.00000

Solution: f([[-0.  0.]]) = 0.00000
```

Figure 3. Convergence and final outcome

Table 2. Comparison of means and standard deviations

	Proposed Algorithm		EPSDE		MPMSDE	
	Mean	Std	Mean	Std	Mean	Std
F1	0	0	0.2278	-0.06017	0.01954	0.04664
F2	-0.18418	-0.0018	-0.18418	0.03751	0.00143	0.0008
F3	-0	0	-0	0	-4.e-05	-5.e-05
F4	0.e+01	0.e+01	-0.e+00	1.e-05	0.e+00	1.e-05

5. CONCLUSIONS

Differential Evolution is one of the famous nature-inspired algorithms dedicated to solving optimization problems especially the problems of multiobjective nature. It has a variety of applications, and it has shown success in solving multidimensional continuous-spaced optimization problems. Much literature was done earlier to enhance this algorithm either by applying different mutation strategies or combining the advantages of best-performing algorithms. In this paper, many DE improvements were reviewed and compared in terms of their used mutation method and overall results to guide future research to enhance such algorithms. Also, a novel variant of DE was developed based on the best features of EPSDE and MPMDE which are some of the latest DE improvements. As a future direction of this work, taking advantage of further best-performing DE variations to create a modified version of the algorithm would be of great interest. Also, incorporating more means of comparison such as disadvantages could enhance this work. In future, the proposed algorithm may be investigated for the real life problems involving multiobjective nature. Moreover, the proposed algorithm being global optimizer may be hybridized with some local optimizer for sake of fine tuning the optimization results in the umbrella of memetic computing. Similarly, other hybrid intelligent techniques can also be investigated.

REFERENCES

[1] Rahman, A. (2019). Memetic computing based numerical solution to Troesch problem. Journal of Intelligent & Fuzzy Systems, 37(1): 1545-1554. https://doi.org/10.3233/JIFS-18579

[2] Rahman, A. (2019). Optimum information embedding in digital watermarking. Journal of Intelligent & Fuzzy Systems, 37(1): 553-564. https://doi.org/10.3233/JIFS-162405

[3] Dash, S., Luhach, A.K., Chilamkurti, N., Baek, S., Nam, Y. (2019). A neuro-fuzzy approach for user behaviour classification and prediction. Journal of Cloud Computing, 8(1): 1-15. https://doi.org/10.1186/s13677-019-0144-9

[4] Dash, S., Tripathy, B.K. (2017). Handbook of research on modeling, analysis, and application of nature-inspired metaheuristic algorithms. IGI Global.

[5] ur Rahman, A. (2018). Applications of hybrid intelligent systems in adaptive communication. In Handbook of Research on Modeling, Analysis, and Application of Nature-Inspired Metaheuristic Algorithms, pp. 183-217. https://doi.org/10.4018/978-1-5225-2857-9.ch010

[6] Azam, M., Sultan, K., Dash, S., Khan, S.N., Khan, M.A.A. (2018). Automated testcase generation and prioritization using GA and FRBS. In International Conference on Advanced Informatics for Computing Research, pp. 571-584. https://doi.org/10.1007/978-981-13-3140-4_52

[7] Atta-ur-Rahman, I.M.Q., Malik, A.N. (2012). Adaptive resource allocation in OFDM systems using GA and fuzzy rule base system. World Applied Sciences Journal, 18(6): 836-844. https://doi.org/10.5829/idosi.wasj.2012.18.06.906

[8] Qureshi, I.M., Malik, A.N., Naseem, M.T. (2014). Dynamic resource allocation in OFDM systems using DE and FRBS. Journal of Intelligent & Fuzzy Systems, 26(4): 2035-2046. https://doi.org/10.3233/IFS-130880

[9] Rahman, A.U., Dash, S., Luhach, A.K. (2021). Dynamic MODCOD and power allocation in DVB-S2: A hybrid intelligent approach. Telecommunication Systems, 76(1): 49-61. https://doi.org/10.1007/s11235-020-00700-x

[10] Rahman, A. (2020). GRBF-NN based ambient aware realtime adaptive communication in DVB-S2. Journal of Ambient Intelligence and Humanized Computing, 1-11. https://doi.org/10.1007/s12652-020-02174-w

[11] Sultan, K., ur-Rahman, A., Zafar, B.A., Aldhafferi, N., Alqahtani, A. (2018). Intelligent multiple relay selection and transmit power-saving with ABC optimization for underlay relay-assisted CRNs. Journal of Communications, 13(7): 377-384. https://doi.org/10.12720/jcm.13.7.377-384

[12] ur-Rahman, A., Qureshi, I.M. (2013). Optimum resource allocation in OFDM systems using FRBS and particle swarm optimization. Nature and Biologically Inspired Computing (NaBIC'13), North Dakota, USA, pp. 174-180.

[13] 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. https://doi.org/10.1023/a:1008202821328

[14] Dash, S., Abraham, A. (2018). Kernel based chaotic firefly algorithm for diagnosing Parkinson's disease. In International Conference on Hybrid Intelligent Systems, pp. 176-188. https://doi.org/10.1007/978-3-030-14347-3_18

[15] Li, X., Wang, L., Jiang, Q., Li, N. (2021). Differential evolution algorithm with multi-population cooperation and multi-strategy integration. Neurocomputing, 421: 285-302. https://doi.org/10.1016/j.neucom.2020.09.007

[16] Liu, J., Lampinen, J. (2002). A fuzzy adaptive differential evolution algorithm. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering. TENCOM '02. Proceedings., pp. 606-611. https://doi.org/10.1109/TENCON.2002.1181348

[17] Qin, A.K., Huang, V.L., Suganthan, P.N. (2009a). Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Transactions on Evolutionary Computation, 13(2): 398-417. https://doi.org/10.1109/TEVC.2008.927706

[18] Guo, Z.Y., Cheng, B., Ye, M., Cao, B.G. (2006). Self-adaptive chaos differential evolution. International Conference on Natural Computation-ICNC 2006: Advances in Natural Computation, pp 972-975. https://doi.org/10.1007/11881070_128

[19] Noman, N., Bollegala, D., Iba, H. (2011). An adaptive differential evolution algorithm. In 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, pp. 2229-2236. https://doi.org/10.1109/CEC.2011.5949891

[20] Atta-ur-Rahman, Qureshi, I.M., Salam, M.H., Muzaffar, M.Z. (2013). Adaptive communication using softcomputing techniques. In International Conference on Soft Computing and Pattern Recognition (SoCPaR), Hanoi, Vietnam, pp. 19-24.

[21] Zhang, J., Sanderson, A.C. (2009). JADE: Adaptive differential evolution with optional external archive. IEEE Transactions on Evolutionary Computation, 13(5): 945-958. https://doi.org/10.1109/TEVC.2009.2014613

[22] Ronkkonen, J., Kukkonen, S., Price, K.V. (2005). Real-parameter optimization with differential evolution. In 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK, pp. 506-513. https://doi.org/10.1109/CEC.2005.1554725

[23] Wu, L., Wang, Y., Yuan, X. (2011). Design of 2-D recursive filters using self-adaptive mutation differential evolution algorithm. International Journal of Computational Intelligence Systems, 4(4): 644-654. https://doi.org/10.1080/18756891.2011.9727819

[24] Wang, B.C., Li, H.X., Li, J.P., Wang, Y. (2018). Composite differential evolution for constrained evolutionary optimization. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(7): 1482-1495. https://doi.org/10.1109/TSMC.2018.2807785

[25] 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 Journal, 11(2): 1679-1696.

https://doi.org/10.1016/j.asoc.2010.04.024
[26] Fan, Q., Yan, X. (2015). Self-adaptive differential evolution algorithm with discrete mutation control parameters. Expert Systems with Applications, 42(3): 1551-1572. https://doi.org/10.1016/j.eswa.2014.09.046
[27] Mezura-Montes, E., Velázquez-Reyes, J., Coello, C.C. (2006). Modified differential evolution for constrained optimization. In 2006 IEEE International Conference on Evolutionary Computation, Vancouver, BC, Canada, pp. 25-32. https://doi.org/10.1109/CEC.2006.1688286
[28] Zou, D., Wu, J., Gao, L., Li, S. (2013). A modified differential evolution algorithm for unconstrained optimization problems. Neurocomputing, 120: 469-481. https://doi.org/10.1016/j.neucom.2013.04.036
[29] Wu, G., Mallipeddi, R., Suganthan, P.N., Wang, R., Chen, H. (2016). Differential evolution with multi-population based ensemble of mutation strategies. Information Sciences, 329(1): 329-345. https://doi.org/10.1016/j.ins.2015.09.009
[30] Li, X., Yin, M. (2016). Modified differential evolution with self-adaptive parameters method. Journal of Combinatorial Optimization, 31: 546-576. https://doi.org/10.1007/s10878-014-9773-6
[31] 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(2): 1679-1696. https://doi.org/10.1016/j.asoc.2010.04.024
[32] Wang, S., Li, Y., Yang, H. (2017). Self-adaptive differential evolution algorithm with improved mutation mode. Applied Intelligence, 47(3): 644-658. https://doi.org/10.1007/s10489-017-0914-3
[33] Sun, G., Xu, G., Jiang, N. (2020). A simple differential evolution with time-varying strategy for continuous optimization. Soft Computing, 24(4): 2727-2747. https://doi.org/10.1007/s00500-019-04159-0
[34] Sun, G., Yang, B., Yang, Z., Xu, G. (2020). An adaptive differential evolution with combined strategy for global numerical optimization. Soft Computing, 24(9): 6277-6296. https://doi.org/10.1007/s00500-019-03934-3
[35] Ali, I.M., Essam, D., Kasmarik, K. (2021). Novel binary differential evolution algorithm for knapsack problems. Information Sciences, 542(1): 177-194. https://doi.org/10.1016/j.ins.2020.07.013
[36] Qureshi, I.M., Naseem, M.T., Muzaffar, M.Z. (2012). A GA-FRBS based rate enhancement scheme for OFDM based hyperlans. In 2012 10th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, pp. 153-158. https://doi.org/10.1109/FIT.2012.35
[37] Qureshi, I.M., Naseem, M.T. (2012). Adaptive resource allocation for OFDM systems using fuzzy rule base system water-filling principle and product codes. In 2012 12th International Conference on Intelligent Systems Design and Applications (ISDA), Kochi, India, pp. 805-810. https://doi.org/10.1109/ISDA.2012.6416640
[38] Qureshi, I.M., Salam, M.H., Muzaffar, M.Z. (2013). Adaptive communication using softcomputing techniques. In 2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR), Hanoi, Vietnam, pp. 19-24. https://doi.org/10.1109/SOCPAR.2013.7054131
[39] Qureshi, I.M., Salam, M.H., Naseem, M.T. (2013). Efficient link adaptation in OFDM systems using a hybrid intelligent technique. In 13th International Conference on Hybrid Intelligent Systems (HIS 2013), Gammarth, Tunisia, pp. 12-17. https://doi.org/10.1109/HIS.2013.6920471
[40] Rahman, A., Qureshi, I., Malik, A., Naseem, M. (2014). A real time adaptive resource allocation scheme for OFDM systems using GRBF-neural networks and fuzzy rule base system. International Arab Journal of Information Technology (IAJIT), 11(6): 593-601.
[41] Atta-ur-Rahman, M.A., Zaman, G. (2016). Performance comparison of product codes and cubic product codes using FRBS for robust watermarking. International Journal of Computer Information Systems and Industrial Management Applications (IJCISIMA), 8(1): 57-66.
[42] Rahman, A. (2017). Applications of softcomputing in adaptive communication. International Journal Control Theory and Applications, 10(18): 81-93.
[43] Sultan, K., Aldhafferi, N., Alqahtani, A. (2017). Differential evolution assisted MUD for MC-CDMA systems using non-orthogonal spreading codes. In International Conference on Innovations in Bio-Inspired Computing and Applications, pp. 36-48.
[44] Muzaffar, M.Z., Qureshi, I.M., Rahman, A., Alhaidari, F.A., Khan, M.A.A. (2018). Compressed sensing for security and payload enhancement in digital audio steganography. Journal of Information Hiding and Multimedia Signal Processing, 15(6): 1506-1517.
[45] ur-Rahman, A. (2018). Efficient decision based spectrum mobility scheme for cognitive radio based v2v communication system. Journal of Communications, 13(9): 498-504. https://doi.org/10.12720/jcm.13.9.498-504
[46] Ahmed, M.I.B. (2019). Virtual Clinic: A CDSS Assisted Telemedicine Framework. In Telemedicine Technologies, pp. 227-238.
[47] Rahman, A.U., Alqahtani, A., Aldhafferi, N., Nasir, M.U., Khan, M.F., Khan, MA., Mosavi, A. (2022). Histopathologic oral cancer prediction using oral squamous cell carcinoma biopsy empowered with transfer learning. Sensors, 22(10): 3833. https://doi.org/10.3390/s22103833
[48] Rahman, A.U., Abbas, S., Gollapalli, M., Ahmed, R., Aftab, S., Ahmad, M., Khan, M.A., Mosavi, A. (2022). Rainfall prediction system using machine learning fusion for smart cities. Sensors, 22(9): 3504. https://doi.org/10.3390/s22093504
[49] Ibrahim, N.M., Gabr, D.G.I., Rahman, A.U., Dash, S., Nayyar, A. (2022). A deep learning approach to intelligent fruit identification and family classification. Multimedia Tools and Applications, 1-16. https://doi.org/10.1007/s11042-022-12942-9
[50] Gollapalli, M., Rahman, A.U., Musleh, D., Ibrahim, N.M., Khan, M.A., Abbas, S., Atta, A., Khan, M.A., Farooqui, M., Ahmed, M.I.B. (2022). A neuro-fuzzy approach to road traffic congestion prediction. Computers, Materials and Continua, 73(1): 295-310. http://dx.doi.org/10.32604/cmc.2022.027925
[51] Mahmud, M., Lee, M., Choi, J.Y. (2020). Evolutionary-based image encryption using RNA codons truth table. Optics & Laser Technology, 121: 105818. https://doi.org/10.1016/j.optlastec.2019.105818
[52] Alhaidari, F., Shaib, N.A., Alsafi, M., Alharbi, H., Alawami, M., Aljindan, R., Rahman, A.U., Zagrouba, R. (2022). ZeVigilante: Detecting zero-day malware using

machine learning and sandboxing analysis techniques. Computational Intelligence and Neuroscience, 2022: 1615528. https://doi.org/10.1155/2022/1615528
[53] Ghazal, T.M., Al Hamadi, H., Nasir, M.U., Rahman, A., Gollapalli, M., Zubair, M., Khan, M.A., Yeob Yeun, C. (2022). Supervised machine learning empowered multifactorial genetic inheritance disorder prediction. Computational Intelligence and Neuroscience, 2022: 1051388. https://doi.org/10.1155/2022/1051388
[54] Alotaibi, S.M., Basheer, M.I., Khan, M.A. (2021). Ensemble machine learning based identification of pediatric epilepsy. Computers, Materials & Continua, 68(1): 149-165.
[55] Alhaidari, F., Almotiri, S.H., Al Ghamdi, M.A., Khan, M.A., Rehman, A., Abbas, S., Khan, K.M. (2021). Intelligent software-defined network for cognitive routing optimization using deep extreme learning machine approach. Computers, Materials & Continua, 67(1): 1269-1285.
[56] Ahmed, M.I.B., Alotaibi, S., Rahman, A.U. et al. (2022). A review on machine learning approaches in Identification of pediatric epilepsy. SN COMPUT. SCI. 3(437). https://doi.org/10.1007/s42979-022-01358-9.
[57] Rahman, A., Asif, R.N., Sultan, K., Alsaif, S.A., Abbas S., Khan, M.A., Mosavi, A. (2022). ECG classification for detecting ECG arrhythmia empowered with deep learning approaches. Computational Intelligence and Neuroscience, 2022: 6852845. https://doi.org/10.1155/2022/6852845
[58] Rahman, A., Nasir, M.U., Gollapalli, M, Alsaif, S.A., Almadhor, A.S., Mehmood, S., Khan, M.A., Mosavi, A. (2022). IoMT-based mitochondrial and multifactorial genetic inheritance disorder prediction using machine learning. Computational Intelligence and Neuroscience, 2022: 2650742. https://doi.org/10.1155/2022/2650742
[59] Rahman, A., Nasir, M.U., Gollapalli, M., Zubair, M. et al. (2022). Advance genome disorder prediction model empowered with deep learning. IEEE, 10: 70317-70328.
[60] Rahman, A., Musleh, D., Nabil, M., Alubaidan, H., Gollapalli, M., Krishnasamy, G., Almoqbil, D., Khan, M.A.A., Farooqui, M., Ahmed, M.I.B., Ahmed, M.S., Mahmud, M. (2022). Assessment of information extraction techniques, models and systems. Mathematical Modelling of Engineering Problems, 9(3): 683-696. https://doi.org/10.18280/mmep.090315