INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Design Optimization Using Modified Differential Evolution Algorithm

Pooja Tiwari[1], **Vishnu Narayan Mishra**[1]*, **Raghav Prasad Parouha**[1]

1 Department of Mathematics, Indira Gandhi National Tribal University, , Amarkantak, 484887, Madhya Pradesh, India

Check for updates

OPEN ACCESS

Received: 06-06-2023

Accepted: 09-10-2023

Published: 17-11-2023

Citation: Tiwari P, Mishra VN, Parouha RP (2023) Design Optimization Using Modified Differential Evolution Algorithm. Indian Journal of Science and Technology 16(43): 3917-3926. https://doi.org/10.17485/IJST/v16i43.1392

* **Corresponding author**.

vishnunarayanmishra@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: The key objective of this article is to suggest a modified differential evolution (MDE) algorithm for design problem optimization particularly reactor network design (RND) problem. **Methods:** During the few decades differential evolution (DE) algorithm achieved noticeable progress and solved a wide variety of optimization issues. However, the DE suffers from low diversification, poor exploration ability and stagnation. Hence, using concept of the particle swarm optimization mechanism (PSO), suggested MDE employed new mutation operator, to balance exploitation and exploration activities. Also, on the basis of time-varying scheme new mutation operator integrates new control parameter, to avoid stagnation. A group of 6 unconstrained benchmark functions are solved, to investigate the presentation of MDE algorithm. Moreover, its practical superiority is further verified by solving RND problem. **Findings:** The experiential results show that the suggested MDE performs well in each case of unconstrained benchmark functions with the highest rate of success. Moreover, optimize the RND problem very effectively with the lowest time (2.98s) and fewer number of function evaluations (12729). Furthermore, outcomes suggest that the proposed MDE exhibits a better or at least competitive performance compared to evolutionary algorithms. **Novelty:** The exploitation and exploration ability of the suggested MDE are balanced efficiently due to use of memory facts (i.e. novel mutation operator) and adapted (i.e. new time-varying) control parameters.

Keywords: Evolutionary algorithms; differential evolution; mutation; control parameter; design optimization

1 Introduction

Optimization has remained a hot spot of research in several areas for instance engineering design optimization problems [(1)], trajectory design optimization problems [(2–4)], multi-objective problems [(5)] etc. Each optimization issue is defined by its objective function, constraints, and decision variables. Also, the optimization process aims to identify the best solution among all available options. During the two decades, a variety of algorithms have been raised for design optimization problems. These algorithms are roughly classified into two broad categories: mathematical programming methods (MPMs) and meta-heuristic algorithms (MAs). In general, MPMs can rapidly converge toward a single optimal solution and usually suffer from low efficiency and are sometimes infeasible [6]. Thus, as a class of population-based algorithms, MAs are broadly employed to tackle challenging design optimization problems. These algorithms can be categorized based on their underlying concepts into evolutionary algorithms (drawing inspiration from natural evolutionary principles), physics-inspired algorithms (insights from physical phenomena), human behavior-based algorithms (stem from social learning in human societies), and swarm intelligence algorithms (inspiration from social behaviors in biological populations) [7]. Some most prominent and/or State-of-the-art (SOTA) MAs developed in literature are Genetic algorithm [8], Differential evolution algorithm [9], Particle swarm optimization [10], Equilibrium optimizer [11], Stock exchange trading optimization algorithm [12] and so on.

Among many MAs, Differential evolution (DE) algorithm has materialized as one of the best widespread algorithm since its initiation in 1995 [9]. It attained manifest growth in the past two decades [13]. Also, in consequence of its easy execution and efficiency it has productively solved various complex optimization issues for example structural optimization [14], optimal designs in the chemical sciences [15] and others. But, similar to other MAs, DE has been reported to encounter drawbacks including falls into local minima and how to balance the exploitation and exploration capability. To overcome these drawbacks, substantial works has been described in the level of algorithm, and control parameters evolution individually. Some of the related work are studied and mentioned as follows.

(i). DE/current-to-pbest/1-X [16] - it is a revised form of widely used mutation operator DE/current-to-pbest/1.

(ii). Hip-DE [17] - it is based on historical population based mutation operator and novel adaptive mutant crossover parameter mechanisms.

(iii). TPDE [18] - it used tri-population based mechanism and 3 sets of adaptive control parameters (F & CR).

(iv). DE/current-to-better/1 [19] - it is an innovative version of the mutation operation.

(v). bDE-MsAC [20] - it utilized 5 modified mutation and multi-mutation strategies autonomy.

(vi). EFDE [21] - it utilized adaptive mutation strategy and control parameters.

(vii). DE/current-to-best/2 [22] – it used a new mutation variants and a self-adapted crossover procedure.

(viii). SCJADE [23] - it employed modified control parameter.

(ix). SIDE [24] - it utilized superior-inferior mutation and dynamic control factor adjustment strategy.

(x). ADEDMR [25] - it used deeply-informed mutation ("DE/target-pbest/1/bin") and restart mechanism.

(xi). AMSC [26] - it utilized auxiliary mutation strategy into split subpopulation.

As a result, all above mentioned the DE variants (used evolution at the level of mutation operators and control parameters) have been reported to encounter drawbacks including falls into local minima and how to balance the exploitation and exploration capability. But they are failed to overcome these limitations due to own pros and cons. Hence, there is a need to propose a novel/modified variant of DE which may resolve its desired requirements.

1.1 Inspiration/motivation/research gaps

The motivations of the proposed framework lie in the following facts.

(i). $v_{i,j}^t = x_{r_1}^t + F \times (x_{r_2}^t - x_{r_3}^t)$ is extensively used mutation scheme and effectively balanced population diversity [9]. In contrast, it has slow convergence rate [13].

(ii). $v_{i,j}^t = x_{r_1}^t + F \times (x_{r_2}^t - x_{r_3}^t) + F \times (x_{r_4}^t - x_{r_5}^t)$ has enhanced perturbation than $v_{i,j}^t = x_{r_1}^t + F \times \left(x_{r_2}^t - x_{r_3}^t\right)$. But, it may fail to provide exploitation facility during the search evolution [13].

(iii). $v_{i,j}^t = x_{best} + F \times \left(x_{r_1}^t - x_{r_2}^t\right) + F \times \left(x_{r_3}^t - x_{r_4}^t\right)$, $v_{i,j}^t = x_{i,j}^t + F \times \left(x_{r_1}^t - x_{r_2}^t\right)$ and $v_{i,j}^t = x_{i,j}^t + F \times \left(x_{best} - x_{i,j}^t\right) + F \times \left(x_{r_1}^t - x_{r_2}^t\right)$ has better exploitation ability. But, they have low exploration capability when solving multimodal optimization problems [19,20].

(iv). Various mutation schemes presented in the literature [13], to decrease the DE disadvantages. But, want essential refinement to enhance the DE search capability [20].

(v). DE might be not stanching the previous best memory/vector information in the evolution process. Hence, it may loss of the best vectors and leads to premature convergence [27].

(vi). The no-free-lunch (NFL) theorem [28] directs that no single algorithms can solve all real-life issues. It motivates to develop new EAs which perform well for complex optimization issues.

1.2 Major contribution

Motivated by the above arguments, this study suggested a modified DE, called MDE, for design optimization issues. Contribution of the present work is listed as below.

(i). Brief and effective survey on recent-past DE's, to find aforesaid research gaps and inspiration.

(ii). MDE adopts novel mutation scheme, to balance exploitation and exploration capability.

(iii). MDE introducing time-varying/self-adaptive/dynamic varying mutant control parameters, to accelerate the convergence.

(iv). MDE used to solve complex unconstrained benchmark functions and reactor network design (RND) optimization problem, to show the superiority, efficiency and effect over other techniques.

The article rest part is prepared as - Section 2 illustrates the suggested MDE. Section 3 provides the validation of MDE on unconstrained benchmark suites. In Section 4, suggested MDE used to solve reactor network design (RND) optimization problem. Conclusions of the whole article and future plans reported in Section 5.

2 Proposed modified DE (MDE)

In this section classic DE outlined firstly then explained suggested MDE in detail.

2.1 Classic DE

DE is a population centered stochastic optimizer which has similar steps with other evolution methods, namely initialization, mutation, crossover and selection. In each cycle mutation, crossover to selection operator used in the evolution process and form new solution vectors. Mutation, crossover and selection cycles are repeated up-to predefined stopping conditions. Following are the implementation steps of the classic DE.

I. Initialization

Aimed at D-dimensional problem optimization, a group of random sampling points (target vectors) $x_{i,j}^{t} = (x_{i,1},\ x_{i,2},\ \ldots, x_{i,D}) i = 1,\ 2,\ \ldots,\ NP$ and $j = 1,\ 2,\ \ldots,\ D$ called the population initialization (NP- population size and D - dimension) is generated randomly in specified limits, at 't^{th}' iteration. NP size initial poputational generated randomly by using following equations.

$$x_{i,j}^{t} = x_i^{min} + rnd\,(0,\ 1)\left(x_i^{max} - x_i^{min}\right) \tag{1}$$

where $i = 1,\ \ldots,\ NP,\ j = 1,\ \ldots,\ D$, t = iteration number, x_i^{min} & x_i^{max} = minimum and maximum value of i^{th} variable.

II. Mutation

$v_{i,j}^{t} = (v_{i,1},\ v_{i,2},\ \ldots, v_{i,D})$ called mutant vector is formed as

$$v_{i,j}^{t} = x_{r_1}^{t} + F \times \left(x_{r_2}^{t} - x_{r_3}^{t}\right) \tag{2}$$

where x_{r_1}, x_{r_2} *and* $x_{r_3} \in (1,\ NP]$, $r_1 \neq r_2 \neq r_3 \neq i$ and $F \in\ (0,\ 1]$ is specified as mutant factor.

III. Crossover

$u_{i,j}^{t} = (u_{i,1},\ u_{i,2},\ \ldots, u_{i,D})$ called trial vector is formed as

$$u_{i,j}^{t} = \begin{cases} v_{i,j}^{t}; if\ rnd \leq CR \\ x_{i,j}^{t}; Otherwise \end{cases} \tag{3}$$

where rnd = uniformly random number spread among 0 and 1, $CR \in (0,\ 1]$ is indicated as crossover constant.

IV. Selection

It is formed as below.

$$x_{i,j}^{t+1} = \begin{cases} u_{i,j}^{t}; if\ f(u_{i,j}^{t}) \leq f(x_{i,j}^{t}) \\ x_{i,j}^{t}; Otherwise \end{cases} \tag{4}$$

V. Termination

Repeats II-V else stopped as per criteria of termination.

2.2 Modified DE (MDE)

Encouraged by literature investigation a modified DE (MDE) presented in this paper, to decrease DE disadvantages. The suggested MDE is dissimilar for DE in the aspect of novel mutation strategy with its control parameter. It illustrated as follows.

Using the concept of particle swarm optimization (PSO) [10], $v_{i,j}^t$ i.e. mutation vectors created as follows.

$$v_{i,j}^t = x_{i,j}^t + F_1 \times \left(xbest_{i,j}^t - x_{i,j}^t\right) + F_2 \times \left(xbetter_j^t - x_{i,j}^t\right) + F_3 \times \left(xworst_j^t - x_{i,j}^t\right) \tag{5}$$

where- $xbest_{i,j}^t$ = best vectors, $xbetter_j^t$ = better vectors, and $xworst_j^t$ = worst vectors. These vectors are restructured as follows.

$$xbest_{i,j}^t = \begin{cases} x_{i,j}^t;\ if\ (x_{i,j}^t) < f(xbest_{i,j}^{t-1}) \\ xbest_{i,j}^{t-1};if\ (x_{i,j}^t) \geq f(xbest_{i,j}^{t-1}) \end{cases}$$

$xbetter_j^t = minimum\{xbest_{i,j}^t\}$ & $xworst_j^t = maximum\{xbest_{i,j}^t\}$

Moreover, F_1, F_2 & F_3 are the novel control parameters defined as follows.

$F_1 = \left(\frac{t-1}{t_{max}-1}\right) \times F_{1,\,initial} - \left(F_{1,\,final} - F_{1,\,initial}\right)$

$F_2 = \left(\frac{t-1}{t_{max}-1}\right) \times\ F_{2,\,initial} - \left(F_{2,\,final} - F_{2,\,initial}\right)$ & $F_3 = (1 - exp\,(F_2 \times t)) \times F_1$

where t_{max} and t = maximum and current iteration number.

Moreover, mutant factors (F_1, F_2 & F_3) have the subsequent quality, throughout the search procedure.

(i). F_1 initiate with big value and gradually falls to a small value, while F_2 initiate with small value and gradually upturns to a large value. In earlier period, large F_1 and small F_2 values are allowed vectors to travel freely over the search space, in place of affecting to the population finest. Likewise, small F_1 and large F_2 values are indorsed vectors to converge the global best, in later phase.

(ii). F_3 quickly upsurge in earlier period then gradually shrinkage in latter period. It supports the vectors to find suitable direction and better movement position.

After a wide investigation, $F_{1,initial} = F_{2,final} = 2.5$, $F_{1,final} = F_{2,initial} = 0.5$ are fixed for MDE for entire experiments. The F_1, F_2, and F_3 variation as per iteration number are depicted in Figure 1.

Overall, in suggested modified DE (MDE) - a new mutation scheme, using the perception of PSO, used to trade off the exploitation and exploration as well as new time-varying mutant control parameters incorporated in the suggested mutation scheme, to escaping local optima and keep evolving. Using the features of memory and robustly altered control parameters, exploitation and exploration ability of MDE may well-balanced. Also, an admitted feature of MDE follows to speeding up convergence significantly. The steps of the MDE are same as DE, instead of mutation scheme (i.e. Equation (5) used in place of Equation (2) at the time of evolution process).

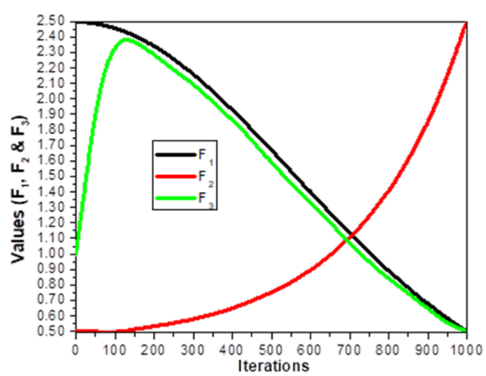


Fig 1. i, F_2 & F_3 **variation as per number of iteration**

Implementation steps of MDE

Step 1. Initialization

initialized MDE parameters and create initial population.

Step 2. Start iteration/evolution

WHILE the termination norm is not fulfilled

DO

Step 2.1: compute mutation as per Equation (5)

Step 2.2: compute crossover as per Equation (3)

Step 2.3: update population through selection Equation (4)

$$t = t + 1$$

End WHILE

3 3. Validation of suggested MDE

Efficiency of developed MDE is thoroughly checked in 6 complex unconstrained benchmark functions. Particulars of these benchmark functions specified in Table 1. Simulation results experimented on Core(TM) Intel(R) i7-7200U CPU, 2.50GHz, 16GB RAM, MATLAB R2021a software with Windows 10 (64-bit) operating system. To conduct a fair evaluation- 30 population size, 10000 maximum iterations and 30 trail runs set for developed MDE which is same as comparative methods. The other parameter settings of MDE mentioned in above section and other methods can be found in the respective papers. The boldface in each table reveals the best results. The experimental results of developed MDE with the other methods presented as follows.

Table 1. Unconstrained benchmark function

Function name	Properties		
	Trait	**Search range**	f_{min}
Sphere $F_1(x) = \sum_{i=1}^{d} x_i^2$	Unimodal	[-100, 100]	0
Schwefel $F_2(x) = \sum_{i=1}^{D} (x\| + \prod_{i=1}^{D} \|x_i\|$	Unimodal	$[-10, 10]$	0
Rosenbrock $F_3(x) = \sum_{i=1}^{D-1} \left((x_i - 1)^2 + 100 \left(x_{i+1} - x_i^2 \right)^2 \right)$	Unimodal	[-30, 30]	0
Rastrigin $F_4(x) = \sum_i^D \left(x_i^2 - 10cos(2\pi xi) + 10 \right]$	Multimodal	$[-5.12, 5.12]$	0
Ackley $F_5(x) = 20 + e - 20e^{-\left(\frac{1}{5} \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} \right)} - e^{-\left(\frac{1}{D} \sum_{i=1}^{D} cos(2\pi x_i) \right)}$	Multimodal	$[-32, 32]$	0
Griewank $F_6(x) = \frac{1}{4000} \sum_i^D x_i^2 - \prod_i^D cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	Multimodal	$[-600, 600]$	0

The experimental results of developed MDE on 6 complex unconstrained benchmark functions equated with DE/rand/1 [9], DE/best/1 [29], DE/target-to-best/1 [30], GDE [31] and PSODE [32]. Mean and standard deviation (SD) over 30 trail runs of MDE with other methods are stated in Table 2. It can notice that from Table 2, projected MDE attained the competitive results with respect to all compared methods in all functions. Also, results of Table 2 confirmed that the capability to catch the global optimum of MDE is greater than others. Less SD of MDE on all unconstrained benchmark function indicates its constancy. Further, to measure the experimental results significance of MDE with others SR (success rate) % ($success\ Rate = \frac{number\ of\ successful\ run}{total\ runs}$ if $f(x) - f(x^*) \leq 0.0001$ than a run is stated as a successful run, where $f(x^*)$ and $f(x)$ is the known and obtained optima respectively) and number of function evaluations (FEs) on considered benchmark suites are reported in Table 3. This table shows that, proposed MDE has less number of FEs and highest success rate percentage on each benchmark function compared to others. It illustrates that, MDE has faster convergence ability and highest reliability than other methods.

Table 2. Numerical comparison results on 6 unconstrained benchmark function

Function	Criteria	"DE/rand/1" [9]	"DE/best/1" [29]	"DE/target-to-best/1" [30]	GDE [31]	PSODE [32]	MDE
F_1(x)	Mean	1.35E-03	4.96E-04	1.09E-04	6.07E-24	1.44E-150	**0**
	SD	5.30E-04	3.32E-04	4.72E-05	8.53E-24	5.72E-150	**0**
F_2(x)	Mean	2.13E-01	2.88E-02	2.04E-02	1.75E-07	5.14E-84	**0**
	SD	7.31E-02	7.52E-03	8.34E-03	4.18E-07	1.43E-83	**0**
F_3(x)	Mean	2.48E-02	2.75E-02	2.02E-02	1.89E-02	2.83E-54	**0**
	SD	6.14E-03	6.85E-03	5.10E-03	6.10E-03	3.26E-60	**0**
F_4(x)	Mean	1.96E+02	1.10E+02	2.01E+02	4.74E+01	5.79E-15	**0**
	SD	7.62E+01	1.89E+01	6.94E+00	1.20E+01	1.00E-14	**0**
F_5(x)	Mean	1.79E-02	8.16E-03	3.60E-03	2.12E-10	1.09E-14	**1.01E-15**
	SD	3.40E-03	2.81E-03	9.84E-04	1.12E-10	3.05E-15	**2.51E-16**
F_6(x)	Mean	7.26E-03	5.78E-03	4.03E-03	8.12E-03	1.59E-02	**0**
	SD	2.93E-03	5.36E-03	3.99E-03	9.78E-03	2.39E-02	**0**

Table 3. Statistical comparison on 6 unconstrained benchmark function

Function	Criteria	"DE/rand/1" [9]	"DE/best/1" [29]	"DE/target-to-best/1" [30]	GDE [31]	PSODE [32]	MDE
$F_1(x)$	FEs	118197	112408	91496	72081	18204	**8519**
	SR%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
$F_2(x)$	FEs	115441	109849	91354	66525	15067	**7867**
	SR%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
$F_3(x)$	FEs	102259	103643	87518	74815	16115	**10182**
	SR%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
$F_4(x)$	FEs	99074	98742	127423	53416	7701	**5627**
	SR%	96.70%	**100%**	**100%**	**100%**	**100%**	**100%**
$F_5(x)$	FEs	125543	118926	100000	76646	29757	**17551**
	SR%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
$F_6(x)$	FEs	125777	117946	97213	81422	18394	**9014**
	SR%	60.00%	46.70%	56.70%	**100%**	**100%**	**100%**

Furthermore, the statistical *t*-test [33] results presented in Table 4 , on 6 complex unconstrained benchmark suites. It should be noticed that from this table, most of the *p*-values are below 0.05, which illustrate that convergence of MDE is enhanced successfully. Additionally, the Friedman's ranking test [33] testified on all associated algorithms on 6 complex unconstrained benchmark suites and results described in Table 5. It specifies that, projected MDE reaches the best ranking in all functions.

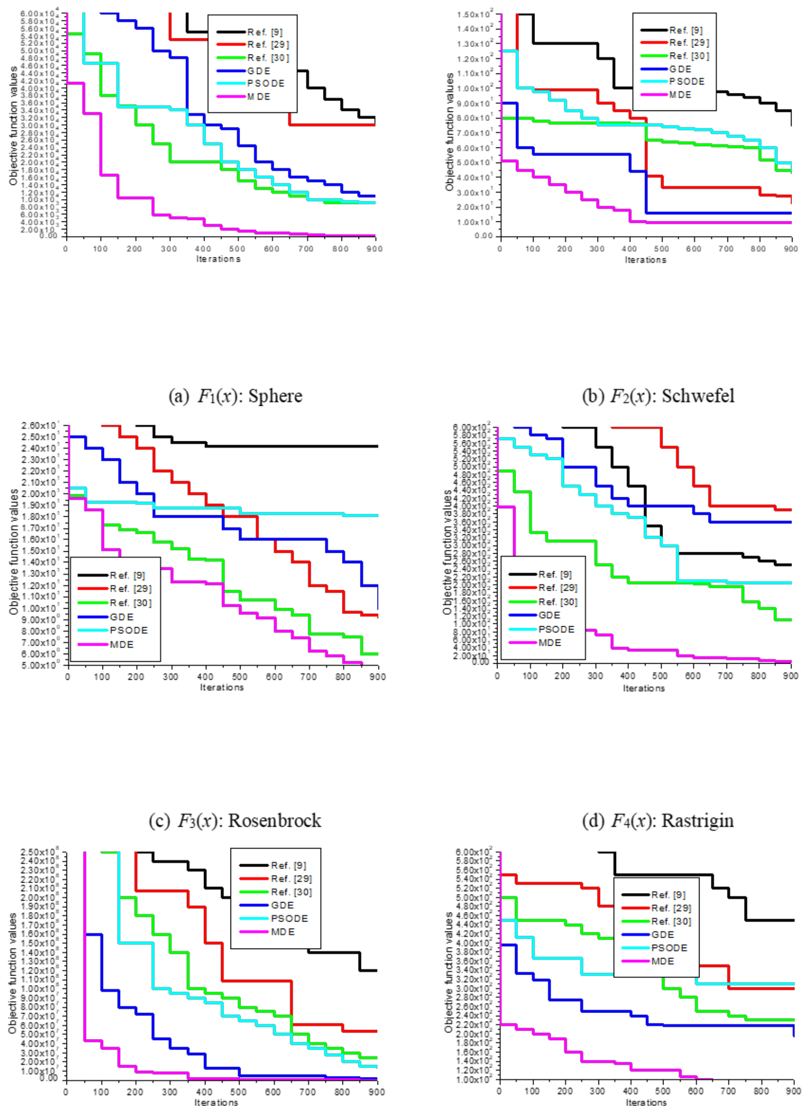
Table 4. The statistical *t* –test (*p*-values) for MDE vs other algorithms

Function	"DE/rand/1" vs MDE	"DE/best/1" vs MDE	"DE/target-to-best/1" vs MDE	GDE vs MDE	PSODE vs MDE
$F_1(x)$	1.21E-16	1.11E-12	1.02E-18	1.04E-08	1.04E-10
$F_2(x)$	2.11E-15	2.01E-10	1.00E-12	4.12E-10	4.11E-12
$F_3(x)$	1.18E-10	1.20E-16	2.51E-10	2.00E-10	2.54E-08
$F_4(x)$	2.01E-18	3.01E-12	1.21E-18	4.00E-08	1.21E-12
$F_5(x)$	2.21E-12	4.07E-10	1.01E-12	2.21E-08	4.04E-18
$F_6(x)$	1.10E-14	2.12E-16	1.12E-10	1.02E-06	1.01E-16

Table 5. Friedman's average ranking test of different methods

Methods	Ranking
MDE	1.07
PSODE	1.21
GDE	2.11
"DE/target-to-best/1"	3.01
"DE/best/1"	4.09
"DE/rand/1"	4.15

Moreover, to analyse the convergence results of MDE, convergence curve is used in this section. It reflects the speed and convergence accuracy of projected and other methods. The convergence curves of developed MDE with other methods on 6 unconstrained benchmark suites displayed in Figure 2**(a-f)**. In these figures, number of iterations is used in x- axis and objective function values gained from each method on same population/seed are used in y- axis. It can be saw that from these figures, MDE has quicker convergence with better precision in most cases. Hence, it can be said that projected MDE has capability to escape from the local minima effectively. Also, the computational time (s) of MDE with other methods depicted in Figure 3 on 6 complex unconstrained benchmark suites through spider chart. It can be noticed that form this figures, MDE provide better results with less time which signify its powerful search performance.



(a) $F_1(x)$: Sphere (b) $F_2(x)$: Schwefel

(c) $F_3(x)$: Rosenbrock (d) $F_4(x)$: Rastrigin

(e) $F_5(x)$: Ackley (f) $F_6(x)$: Griewank

Fig 2. (a-f): Convergence curves of MDE with other methods

4 Design problem optimization

Here suggested MDE used to solve reactor network design problem. The details of this problem mentioned in Ryoo and Sahinidis(34) and depicted in Figure 4.

It designs sequence of two continuous stirred tank reactors (CSTR) where the successive reaction $A \to B \to C$ receipts place. CSTR is a kind of chemical reactor that is broadly castoff in industrial processes to yield pharmaceuticals, chemicals, and other products. Also, its major objective is maximization of the concentration of product B (i.e. $x_4 = C_{B2}$) in the exit system.

Mathematically, RND issue represented as follows.

Minimize $f = -x_4$,

subject to – $x_1 + k_1x_2x_5 = 1, x_2 - x_1 + k_2x_2x_6 = 0, x_3 + x_1 + k_3x_3x_5 = 1$

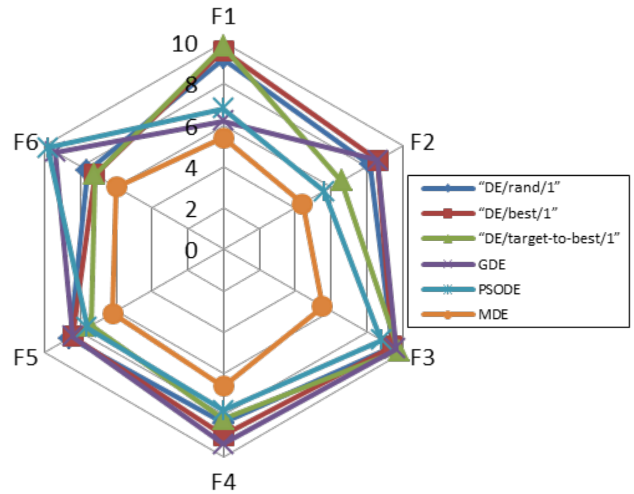


Fig 3. Computational time (s) of MDE with other algorithms

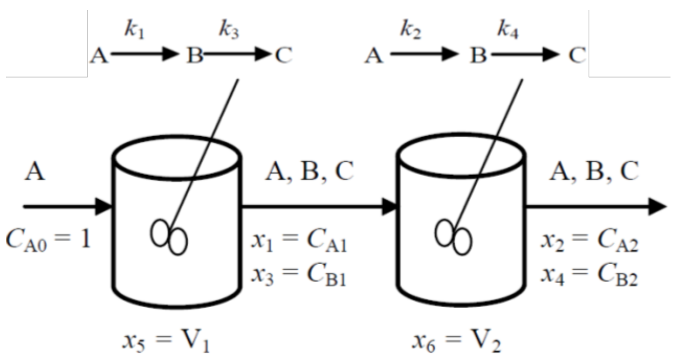


Fig 4. RND problem

$x_4 - x_3 + x_2 - x_1 + k_4 x_4 x_6 = 0, x_5^{0.5} + x_6^{0.5} \leq 4$

$$\left(0,\ 0,\ 0,\ 0,\ 10^{-5},\ 10^{-5}\right) \leq (x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6) \leq (1,\ 1,\ 1,\ 1,\ 16,\ 16)$$

where $k_1 = 0.09755988, k_2 = 0.099k_1, k_3 = 0.0391908, k_4 = 0.09k_3$

Moreover, it can be reformulated as follows after eliminating equality constraint.

$$Maximize\ f = \frac{k_2\, x_6\ (1 + k_3\, x_5) + k_1\, x_5\, (1 + k_2\, x_6)}{(1 + k_1\, x_5)\,(1 + k_2\, x_6)\,(1 + k_3\, x_5)\,(1 + k_4\, x_6)}$$

subject to - $x_5^{0.5} + x_6^{0.5} \leq 4,\ \left(10^{-5},\ 10^{-5}\right) \leq (x_5,\ x_6) \leq (16,\ 16)$

The global optimum value of $f = -0.388812$ existed at

$(x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6) = (0.771462,\ 0.516997,\ 0.204234,\ 0.388812,\ 3.036504,\ 5.096052)$

The following parameters used for this problem for all methods - np = 30, t_{max}= 500 / maximum numbers of functions evaluations (FEs) = 100000 and trails/run = 25. The bracket penalty method used in MDE to handle the constraint of the problems [27]. Rest parameters are same as above. The results achieved by presented MDE for this problem equated with DE [9] and PSO [10]. Table 6 shows the gained results of MDE, DE and PSO in terms of best, mean, median, worst, SD, average time (s) and numbers of FEs over 25 run on RND problem. It can be observed that from this table, MDE has higher capability to find the better solution than others in all criteria. Also, stability and produce quality solution of MDE is better than DE and PSO, as SD found less in each assessment criteria. As well, comparatively less times and numbers of function evaluation shows the better/faster convergence probability of MDE.

Table 6. Comparison results on reactor network design problem

Methods	Criteria						
	best	mean	median	worst	SD	average time (s)	no. of FEs
MDE	**0.388811**	**0.388811**	**0.388811**	**0.388811**	**1.02E-19**	**2.98**	**12729**
DE [9]	**0.388811**	0.388808	0.388807	0.388725	5.31E-06	3.64	14352
PSO [10]	**0.388811**	0.388789	0.388774	0.388456	1.53E-05	4.59	27582

Moreover, error (difference between average and known global optimum cost of objective function value) against iterations for network design problem depicted in Figure 5. It shows that MDE has very less error values i.e. able to find better solution in few of iterations than DE and PSO. This figure also is evident and reveals the fact that projected MDE produce more accurate results which are close to global optimum value than other EAs methods. From the above analysis, it can be said that the suggested MDE is good optimizer for design optimization problems.

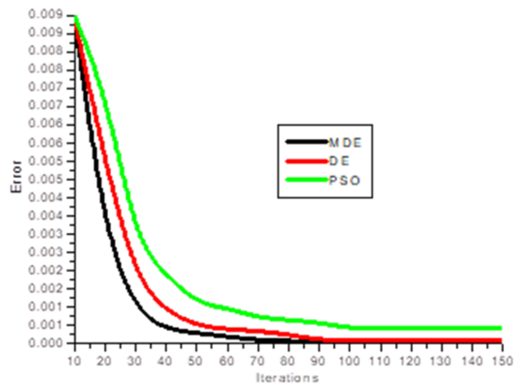


Fig 5. Solution quality measurement of MDE with others on RND problem

5 Conclusion with future works

In this article, a modified differential evolution (MDE) is projected for solving design problem optimization particularly reactor network design (RND) problem. It implemented a novel mutation scheme, created on the perception of particle swarm optimization, which balance population diversity competently. Also, novel time-varying mutant control parameters incorporated with suggested mutation scheme, which helps individuals to escaping from local optima. Using the features of new mutation and robustly altered control parameters, exploitation and exploration ability of MDE is well-adjusted. Also, admitted features of MDE algorithm follows to speeding up convergence competently.

The performance of MDE has verified on 6 multifaceted unconstrained benchmark suites. The experimental results obtained by MDE equated with effective recent-past algorithms. Suggested MDE find the better results efficiently on all benchmark suites with lesser time and highest success rate, due to its new mutation and controlling factors strategies. Furthermore, the proposed MDE used to solve reactor network design (RND) optimization problem. The results produced by MDE are better than all compared algorithms. The proposed MDE produce all results on unconstrained benchmark functions and design optimization problem with less standard deviation, number of function evaluations, and CPU time (s) which shows its stability, reliability, and efficiency to produce best results.

Moreover, suggested MDE algorithm exhibits better convergence, as revealed by the examination of the convergence curves. Hence, advised MDE is a lively alternate of DE for solving design optimization problems. In the future, suggested MDE can be used for real-world multi-objective optimization issues.

References

1) Abualigah L, Elaziz MA, Khasawneh AM, Alshinwan M, Ibrahim RA, Al-Qaness MAA, et al. Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: a comprehensive survey, applications, comparative analysis, and results. *Neural Computing and Applications*. 2022;34(6):4081–4110. Available from: https://doi.org/10.1007/s00521-021-06747-4.

2) Zuo M, Dai G, Peng L, Chen L, Chen X, Song Z. Global optimisation of multiple gravity assist spacecraft trajectories based on search space exploring and PCA. In: 2016 IEEE Congress on Evolutionary Computation (CEC). IEEE. 2016. Available from: https://ieeexplore.ieee.org/document/7744121.
3) Zuo M, Dai G, Peng L, Wang M, Liu Z, Chen C. A case learning-based differential evolution algorithm for global optimization of interplanetary trajectory design. *Applied Soft Computing*. 2020;94:106451. Available from: https://doi.org/10.1016/j.asoc.2020.106451.
4) Zuo M, Dai G, Peng L, Tang Z, Gong D, Wang Q. A differential evolution algorithm with the guided movement for population and its application to interplanetary transfer trajectory design. *Engineering Applications of Artificial Intelligence*. 2022;110:104727. Available from: https://doi.org/10.1016/j.engappai.2022.104727.
5) Zuo M, Gong D, Wang Y, Ye X, Zeng B, Meng F. Process Knowledge-guided Autonomous Evolutionary Optimization for Constrained Multiobjective Problems. *IEEE Transactions on Evolutionary Computation*. 2023;p. 1–1. Available from: https://ieeexplore.ieee.org/document/10040230.
6) Tian Y, Chen H, Ma H, Zhang X, Tan KC, Jin Y. Integrating Conjugate Gradients Into Evolutionary Algorithms for Large-Scale Continuous Multi-Objective Optimization. *IEEE/CAA Journal of Automatica Sinica*. 2022;9(10):1801–1817. Available from: https://ieeexplore.ieee.org/document/9889139.
7) Abualigah L, Diabat A, Mirjalili S, Elaziz MA, Gandomi AH. The Arithmetic Optimization Algorithm. *Computer Methods in Applied Mechanics and Engineering*. 2021;376:1–38. Available from: https://doi.org/10.1016/j.cma.2020.113609.
8) Whitley D. A genetic algorithm tutorial. *Statistics and Computing*. 1994;4(2):65–85. Available from: https://doi.org/10.1007/BF00175354.
9) Storn R, Price K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*. 1997;11:341–359. Available from: https://doi.org/10.1023/A:1008202821328.
10) Eberhart R, Kennedy J. Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks. IEEE. 2002. Available from: https://ieeexplore.ieee.org/document/488968.
11) Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S. Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*. 2020;191:105190. Available from: https://doi.org/10.1016/j.knosys.2019.105190.
12) Emami H. Stock exchange trading optimization algorithm: a human-inspired method for global optimization. *The Journal of Supercomputing*. 2022;78(2):2125–2174. Available from: https://doi.org/10.1007/s11227-021-03943-w.
13) Ahmad MF, Isa NAM, Lim WH, Ang KM. Differential evolution: A recent review based on state-of-the-art works. *Alexandria Engineering Journal*. 2022;61(5):3831–3872. Available from: https://doi.org/10.1016/j.aej.2021.09.013.
14) Georgioudakis M, Plevris V. A Comparative Study of Differential Evolution Variants in Constrained Structural Optimization. *Frontiers in Built Environment*. 2020;6:1–14. Available from: https://doi.org/10.3389/fbuil.2020.00102.
15) Stokes Z, Mandal A, Wong WK. Using Differential Evolution to design optimal experiments. *Chemometrics and Intelligent Laboratory Systems*. 2020;199:103955. Available from: https://doi.org/10.1016/j.chemolab.2020.103955.
16) Zuo M, Dai G, Peng L. A new mutation operator for differential evolution algorithm. *Soft Computing*. 2021;25(21):13595–13615. Available from: https://doi.org/10.1007/s00500-021-06077-6.
17) Meng Z, Yang C. Hip-DE: Historical population based mutation strategy in differential evolution with parameter adaptive mechanism. *Information Sciences*. 2021;562:44–77. Available from: https://doi.org/10.1016/j.ins.2021.01.031.
18) Deng L, Li C, Han R, Zhang L, Qiao L. TPDE: A tri-population differential evolution based on zonal-constraint stepped division mechanism and multiple adaptive guided mutation strategies. *Information Sciences*. 2021;575:22–40. Available from: https://doi.org/10.1016/j.ins.2021.06.035.
19) Mingcheng Z, Guo C. DE/current-to-better/ 1: A new mutation operator to keep population diversity. *Intelligent Systems with Applications*. 2022;14:1–22. Available from: https://doi.org/10.1016/j.iswa.2022.200063.
20) Li Y, Wang S, Liu H, Yang B, Yang H, Zeng M, et al. A backtracking differential evolution with multi-mutation strategies autonomy and collaboration. *Applied Intelligence*. 2022;52(3):3418–3444. Available from: https://doi.org/10.1007/s10489-021-02577-y.
21) Gupta S, Su R. An efficient differential evolution with fitness-based dynamic mutation strategy and control parameters. *Knowledge-Based Systems*. 2022;251:109280. Available from: https://doi.org/10.1016/j.knosys.2022.109280.
22) Fadhil S, Zaher H, Ragaa N, Oun E. A Modified Differential Evolution Algorithm Based on Improving A New Mutation Strategy and Self-Adaptation Crossover. *MethodsX*. 2023;11:1–11. Available from: https://doi.org/10.1016/j.mex.2023.102276.
23) Ghasemi M, Rahimnejad A, Gil M, Akbari E, Gadsden SA. A self-competitive mutation strategy for Differential Evolution algorithms with applications to Proportional–Integral–Derivative controllers and Automatic Voltage Regulator systems. *Decision Analytics Journal*. 2023;7:1–18. Available from: https://doi.org/10.1016/j.dajour.2023.100205.
24) Duan M, Yu C, Wang S, Li B. A differential evolution algorithm with a superior-inferior mutation scheme. *Soft Computing*. 2023;27(23):17657–17686. Available from: https://doi.org/10.1007/s00500-023-09038-3.
25) Zhang Q, Meng Z. Adaptive differential evolution algorithm based on deeply-informed mutation strategy and restart mechanism. *Engineering Applications of Artificial Intelligence*. 2023;126(Part C):107001. Available from: https://doi.org/10.1016/j.engappai.2023.107001.
26) Deng L, Qin Y, Li C, Zhang L. An adaptive mutation strategy correction framework for differential evolution. *Neural Computing and Applications*. 2023;35(15):11161–11182. Available from: https://doi.org/10.1007/s00521-023-08291-9.
27) Parouha RP, Verma P. State-of-the-Art Reviews of Meta-Heuristic Algorithms with Their Novel Proposal for Unconstrained Optimization and Applications. *Archives of Computational Methods in Engineering*. 2021;28(5):4049–4115. Available from: https://doi.org/10.1007/s11831-021-09532-7.
28) Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*. 1997;1(1):67–82. Available from: https://ieeexplore.ieee.org/document/585893.
29) Das S, Abraham A, Chakraborty UK, Konar A. Differential Evolution Using a Neighborhood-Based Mutation Operator. *IEEE Transactions on Evolutionary Computation*. 2009;13(3):526–553. Available from: https://ieeexplore.ieee.org/document/5089881.
30) Cheshmehgaz HR, Desa MI, Wibowo A. Effective local evolutionary searches distributed on an island model solving bi-objective optimization problems. *Applied Intelligence*. 2013;38:331–356. Available from: https://doi.org/10.1007/s10489-012-0375-7.
31) Han MFF, Liao SHH, Chang JYY, Lin CTT. Dynamic group-based differential evolution using a self-adaptive strategy for global optimization problems. *Applied Intelligence*. 2013;39(1):41–56. Available from: https://doi.org/10.1007/s10489-012-0393-5.
32) Liu H, Cai Z, Wang Y. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing*. 2010;10(2):629–640. Available from: https://doi.org/10.1016/j.asoc.2009.08.031.
33) Derrac J, García S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*. 2011;1(1):3–18. Available from: https://doi.org/10.1016/j.swevo.2011.02.002.
34) Ryoo HS, Sahinidis NV. Global optimization of nonconvex NLPs and MINLPs with applications in process design. *Computers & Chemical Engineering*. 1995;19(5):551–566. Available from: https://doi.org/10.1016/0098-1354(94)00097-2.