Journal of Autonomous Intelligence (2024) Volume 7 Issue 3
doi: 10.32629/jai.v7i3.1103

ORIGINAL RESEARCH ARTICLE

Enhancing differential evolution through a modified mutation strategy for unimodal and multimodal problem optimization

Pooja Tiwari*, Vishnu Narayan Mishra, Raghav Prasad Parouha

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

*** Corresponding author:** Pooja Tiwari, pooja.tiwari.igntu@gmail.com

ABSTRACT

Amid a lot of evolutionary methods (EMs), differential evolution (DE) is broadly used for various optimization issues. Though, it has rare shortcomings such as slow convergence, stagnation etc. Likewise, mutation and its control factor choice for DE is extremely inspiring for enhanced optimization. To increase the exploration competence of DE, a modified-DE (M-DE) is advised in this paper. It implemented a new mutation system, thru the perception of particle swarm optimization, to further trade off the population diversity. Meanwhile, centered on time-varying structure, new mutant control parameters incorporated with 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 M-DE is well-adjusted. Also, admitted features of M-DE algorithm follows to speeding up convergence significantly. Finally, to verify the effectiveness of M-DE, groups of assessments have been piloted on six unimodal and seven multimodal benchmark suites. Performance of M-DE compared with different peer DE algorithms. According the investigational results, efficiency of the suggested M-DE technique has been confirmed.

Keywords: evolutionary algorithm; differential evolution; mutation operation; crossover; unimodal and multimodal

ARTICLE INFO

Received: 7 August 2023
Accepted: 7 September 2023
Available online: 8 January 2024

1. Introduction

Currently, vast quantities of evolutionary methods (EMs) are designed to resolve unimodal and multimodal optimization issues[1]. Among several EMs, differential evolution (DE) has appeared as one of the greatest widespread optimizers since its initiation in 1995[2]. It has competence to solve multifaceted optimization issues, due to its easy implementation. Equally, it attained evident improvement in the past two years, owing to its search capacity and proficiency[3]. Besides, the DE effectively applied in several real-life problems, for example power system[4], neural system[5], image analysis[6], chemical manufacturing[7], etc. Conversely, at the time of solving multifaceted optimization problem, the DE faces certain disadvantages for instance stagnation i.e. to escape from local minima[8]. Correspondingly, DE has no possibility to solve all optimization problems efficiently[9].

Henceforth, various innovative DE reforms have been advocated in the literature to increase the DE presentation[10–16]. But, most of the DE variants still face the stagnation problem and provide poor results for complex optimization issues[17]. It happens due to selection of DE mutation and control factors schemes. Hence, various mutation and control factors for DE projected by scholars. For example, Coelho et al.[18] applied the belief space concept of the cultural algorithm as a selection criterion to select between the rand/1 operator and the best/1 operator. Mallipeddi et al.[19] used a group of different mutation and control parameter values during the DE evolution process. Wang et al.[20] used a set of candidates which is generated by diverse mutant vector approaches with randomly choosing control factor from a group of acceptable values respectively. Gong et al.[21] realized two different DE variants by techniques of probability matching and adaptive pursuit respectively. Gong et al.[22] also anticipated a cheap surrogate multi-operative search scheme for DE instead of choosing an operator according to probability. Through this framework, different mutation strategies combination can be easily achieved. Based on multi-population framework, Wu et al.[23] proposed MPEDE. It has 3 mutation approaches, i.e., rand/1, current-to-pbest/1 and current-to-rand/1. Based on probability selection, Zou et al.[24] hybridized rand/2 and mutation rand/1 schemes. The rand/2 selection chance was reduced throughout the evolution process. It achieved better performance than other DEs in small- and medium-scale test cases. Neto et al.[25] suggested self-adaptive DE (SaDE). It used continuous-greedy randomized adaptive search procedure (C-GRASP), to enhance search DE performance. Also, rand/1 or rand/2 mutation operator selected adaptively in SaDE, to create better solution quality over C-GRASP. Zhang et al.[26] used new mutation operators which are selected based on number of improvement failures and quality of each solution. Ma et al.[27] summarized multi-population and/or mutation operator's methods of the DE.

Furthermore, hybridizing DE with local search (LS)[28] and PSO (particle swarm optimization)[29] techniques improves the search performance of the DE. For instance, based on chaotic local (CL) search and a 'shrinking' tactics, Jia et al.[30] offered an active memetic DE. The CLS helped in the early stage to explore a huge search space (it avoids premature convergence) and in the later stage to exploiting a small region (it refines the final solutions). Dhaliwal and Dhillon[31] proposed binary DE (BDE) where binary hill-climbing used as local search techniques. Zuo et al.[32] offered a case learning-based DE by using a local search to solve interplanetary trajectory design optimization problem. Parouha and Verma[33] overviewed DE and PSO developments effectively and suggested there advanced variants and hybrid to solve complex optimization problems. Up-to-date various DE and its hybrids variants suggested for solving complex optimization issues. But they are unable to provide best outcome and falls into local minima; due to unable to used earlier best outcomes[34].

Encouraged by the above literature survey and PSO method, in this paper a modified-DE (M-DE) is presented. It has a new mutation scheme, using the perception of PSO, to trade off the exploration and exploitation. Also, new time-varying mutant control parameters incorporated with 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 M-DE is well-adjusted. Also, admitted features of M-DE algorithm follows to speeding up convergence significantly. Finally, to verify the effectiveness of M-DE, groups of assessments have been piloted on six unimodal and seven multimodal benchmark suites. Presentation of M-DE equated with different peer DE methods. The experimental results of M-DE are better than other compared methods which confirm its efficiency and ability to solve unimodal and multimodal optimization issues.

The article rest part is organized as Section 2 provides classic DE outline. The facts of the suggested M-DE are defined in Section 3. Section 4 shows results with discussions. And Section 5 present conclusion of the whole paper and future plans.

2. Basic DE

Differential Evolution (DE) is a powerful population-based stochastic optimization algorithm that has proven to be effective in solving a wide range of optimization problems. Its underlying principles are similar to other evolutionary methods (EMs), and it follows a multi-step process that iteratively evolves a population of candidate solutions to find the optimal solution. The first step in the classic DE algorithm is the initialization phase. Here, an initial population of candidate solutions, often referred to as individuals or vectors, is generated randomly within the problem's search space. Each individual represents a potential solution to the optimization problem, and the size of the population is determined by a predefined parameter.

Once the initial population is created, the DE algorithm proceeds to the mutation phase. During mutation, three distinct individuals, namely the target vector (denoted as "target"), and two randomly selected vectors from the current population (denoted as "base" and "auxiliary") are combined to form a new trial vector. The trial vector is created by perturbing the "base" vector using the difference between the "auxiliary" vector and the "target" vector, scaled by a mutation parameter called the scaling factor (F). The mutation is a crucial step in DE, as it promotes exploration in the search space.

Following the mutation, the crossover operation is performed. In this phase, each element of the trial vector is compared with a corresponding element in the "target" vector. A random crossover parameter (CR) is used to determine whether the element in the trial vector is retained as part of the new candidate solution. If the randomly generated crossover value is less than CR, the element from the trial vector is selected; otherwise, the element from the "target" vector is retained. This ensures that the new candidate solution retains some characteristics of the original "target" vector, maintaining some level of exploitation. After the mutation and crossover operations are completed, the selection phase follows. In this step, the trial vector is compared to the "target" vector to determine which one becomes a member of the next generation's population. If the trial vector outperforms the "target" vector in terms of fitness (i.e., closer to the optimal solution), it is selected to replace the "target" vector in the next generation. Otherwise, the "target" vector remains unchanged, preserving its position for the next iteration.

The mutation, crossover, and selection cycles are repeated for a predefined number of generations or until a stopping criterion is met. The stopping criterion could be reaching a maximum number of iterations, achieving a satisfactory fitness level, or attaining a certain level of convergence. The strength of DE lies in its ability to effectively balance exploration and exploitation in the search space. The mutation operation promotes exploration by perturbing the candidate solutions, while the crossover and selection operations allow for exploitation by retaining good solutions from the previous generation. This combination of exploration and exploitation enables DE to efficiently converge towards the optimal solution. While the classic DE algorithm has demonstrated its effectiveness in various applications, researchers have proposed numerous modifications and enhancements to address its limitations and adapt it to specific problem domains. Some variants introduce adaptive strategies for adjusting control parameters, dynamic population size, or novel mutation and crossover schemes to improve performance and convergence speed.

In conclusion, Differential Evolution is a versatile and robust optimization algorithm based on evolutionary principles. Its stepwise process of initialization, mutation, crossover, and selection, coupled with the exploration-exploitation balance, has made it a popular choice for solving optimization problems in diverse fields. As researchers continue to explore and innovate in the realm of evolutionary algorithms, DE is likely to remain a significant player in the quest for efficient and effective optimization solutions.

I. Initialization

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

II. Mutation

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

$$v_{i,j}^{t} = x_{r_1}^{t} + F \times \left(x_{r_2}^{t} - x_{r_3}^{t}\right) \tag{1}$$

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

$$u_{i,j}^t = \begin{cases} v_{i,j}^t; \text{if } rnd \le CR \\ x_{i,j}^t; \text{Otherwise} \end{cases} \quad (2)$$

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

$$x_{i,j}^{t+1} = \begin{cases} u_{i,j}^t; \text{ if } f(u_{i,j}^t) \le f(x_{i,j}^t) \\ x_{i,j}^t; \text{ Otherwise} \end{cases} \quad (3)$$

V. Termination

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

3. Suggested modified-DE (M-DE)

In this part, listed the observation from the literature survey (research gaps) then to promote the exploration and exploitation competency balance of DE, a modified-DE (M-DE) is proposed and described with the implementation steps in detailed.

1) $v_{i,j}^t = x_{r_1}^t + F \times (x_{r_2}^t - x_{r_3}^t)$ is extensively used mutation scheme and effectively balanced population diversity[2,35]. In contrast, it has slow convergence rate[35].
2) $v_{i,j}^t = x_{r_1}^t + F \times (x_{r_2}^t - x_{r_3}^t) + F \times (x_{r_4}^t - x_{r_5}^t)$ has enhanced perturbation than $v_{i,j}^t = x_{r_1}^t + F \times (x_{r_2}^t - x_{r_3}^t)$[10,11]. But, it may fail to provide exploitation facility during the search evolution[12,13].
3) $v_i^j(t+1) = x_{best} + F(x_{r_1} - x_{r_2}) v_{i,j}^t = x_{best} + F \times (x_{r_1}^t - x_{r_2}^t) + F \times (x_{r_3}^t - x_{r_4}^t)$, $v_{i,j}^t = x_{i,j}^t + F \times (x_{r_1}^t - x_{r_2}^t)$ and $v_{i,j}^t = x_{i,j}^t + F \times (x_{best} - x_{i,j}^t) + F \times (x_{r_1}^t - x_{r_2}^t)$ has better exploitation ability[14]. But, they have low exploration capability when solving multimodal optimization problems[15,16].
4) Various mutation schemes presented in the literature[19], to decrease the DE disadvantages. Bur, want essential refinement to enhance the DE search capability[36,37].
5) 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[38].

Encouraged by above stated and literature investigation, a modified-DE (M-DE) presented in this article, to overcome the DE disadvantages. The steps of advised M-DE are cited as below.

I. Initialization

In M-DE, NP size initial population generated randomly by using following equations.

$$x_{i,j}^t = x_i^{min} + rnd(0, 1)(x_i^{max} - x_i^{min}) \quad (4)$$

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

Using the concept of PSO[29], $v_{i,j}^t$ i.e. mutation vector created as follows.

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

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

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

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

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

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

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.

1) 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, rather than moving to the population's best. In contrast, small F_1 and large F_2 values are allowed vectors to converge the global best, in latter period.
2) 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 an wide investigation, $F_{1,\text{initial}} = F_{2,\text{final}} = 2.5$, $F_{1,\text{final}} = F_{2,\text{initial}} = 0.5$ are fixed for M-DE for entire experiments. Variation of F_1, F_2, and F_3 according to iteration number are depicted in **Figure 1**.

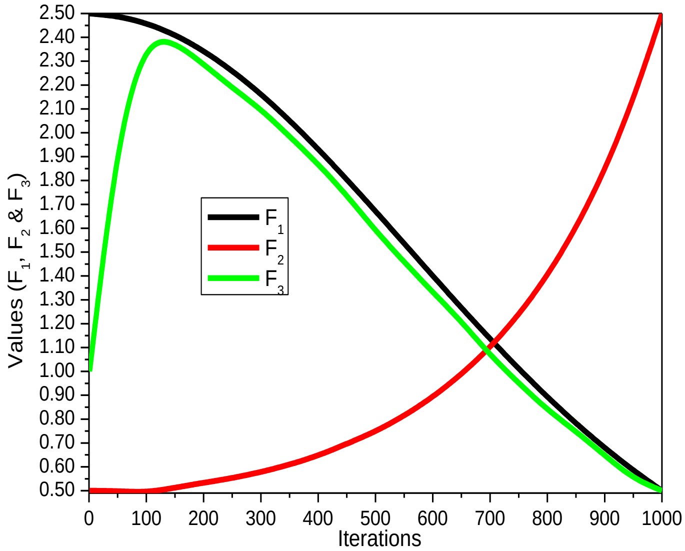


Figure 1. Illustrates how the values of F_1, F_2, and F_3vary with the iteration number during the evolution process.

III. Crossover

Recombines mutant ($v_{i,j}^t$) and target ($x_{i,j}^t$) vector to create trail ($u_{i,j}^t$) vector as follows.

$$u_{i,j}^t = \begin{cases} v_{i,j}^t; \text{if } rnd(0,1) \leq CR \\ x_{i,j}^t \text{ ; if } rnd(0,1) > CR \end{cases} \tag{6}$$

where $rnd(0,1)$ = random number among 0 & 1, CR = crossover rate.

IV. Selection

A greedy selection scheme used in M-DE i.e. if $u_{i,j}^t$ has better or equal function values than it used for next iteration otherwise $x_{i,j}^t$ will used for the next iteration. It works as follows mathematically.

$$x_{i,j}^{t+1} = \begin{cases} u_{i,j}^{t}\ ;\ \text{if}\ f(u_{i,j}^{t}) \le f(x_{i,j}^{t}) \\ x_{i,j}^{t}\ ;\ \text{if}\ f(u_{i,j}^{t}) > f(x_{i,j}^{t}) \end{cases} \tag{7}$$

V. Stopping

Repeat step II–V, else stop as per specified stopping criteria such as t_{max} (maximum iterations).

4. Result and discussion

The developed Modified Differential Evolution (M-DE) algorithm's efficiency was extensively evaluated using six unimodal (F1–F7) and seven multimodal (F8–F13) standard benchmark suites. These benchmark suites are well-known test functions used in the optimization community to assess the performance of various algorithms. **Table 1** provides details about each benchmark suite, including the specific functions used for evaluation. To ensure a fair comparison, the simulation experiments were conducted on a system with an Intel(R) i7-7200U CPU running at 2.50 GHz, 16 GB of RAM, and MATLAB R2021a software on the Windows 10 (64-bit) operating system. The chosen hardware and software configuration are commonly used for optimization research and provide a reliable testing environment.

Table 1. Features of 13 classical benchmark functions.

Functions	Type	Search range	Optimum solution
$F_1 = \sum_{i=1}^{D} x_i^2$	unimodal	[−100,100]	0
$F_2 = \sum_{i=1}^{D}\lvert x\rvert + \prod_{i=1}^{D}\lvert x_i\rvert$		[−10, 10]	0
$F_3 = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2$		[−100,100]	0
$F_4 = max_i\lvert x_i\rvert, 1 \le i \le D$		[−100,100]	0
$F_5 = \sum_{i}^{D}(\lfloor x_i + 0.5\rfloor)^2$		[−30, 30]	0
$F_6 = \left(\sum_{i}^{D} i\, x_i^4\right) + rand[0,1)$		[−100,100]	0
$F_7 = \sum_{i}^{D-1}\left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$		[−1.28, 1.28]	0
$F_8 = \sum_{i}^{D} -x_i sin\left(\sqrt{\lvert x_i\rvert}\right)$	multimodal	[−500,500]	$-418.9829 \times \text{D}$
$F_9 = \sum_{i}^{D}\left[x_i^2 - 10cos(2\pi xi) + 10\right]$		[−5.12, 5.12]	0
$F_{10} = -20exp\left(-\sqrt[0.2]{\frac{1}{D}\sum_{i}^{D} {x_i}^2}\right) - \exp\left(\frac{1}{D}\sum_{i}^{D} cos2\pi xi\right) + 20 + e$		[−32,32]	0
$F_{11} = \frac{1}{4000}\sum_{i}^{D} x_i^2 - \prod_{i}^{D} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$		[−600, 600]	0
$F_{12} = \frac{\pi}{D}\left\{10sin^2(xy_i) + \sum_{i}^{D}(x_i - 1)^2\left(1 + sin^2\left(xy_{i+1)}\right)\right) + (y_D - 1)^2\right\} + \sum_{i}^{D} U(x_i, 10{,}100{,}4)$ where $y_i = 1 + \frac{1}{4}(x_i + 1)$ and $U(x_i, a, k, m) = \begin{cases} k\,(x_i - a)^m, \text{if } x_i > a \\ k\,(-x_i - a)^m, \text{if } x_i < -a \\ 0, \quad \text{otherwise} \end{cases}$		[−50,50]	0
$F13 = 0.1\{sin^2(3\pi x_i)\} + \sum_{i=1}^{D}(x_i - 1)^2\,[1 + sin^2(3\pi x_i + 1)] + (x_D - 1)^2 + \sum_{i}^{D} U(x_i, 5, 100{,}4)$ where $U(x_i, a, k, m) = \begin{cases} k\,(x_i - a)^m, \text{if } x_i > a \\ k\,(-x_i - a)^m, \text{if } x_i < -a \\ 0, \quad \text{otherwise} \end{cases}$		[−50,50]	0

For the evaluation, several parameter settings were standardized across all experiments. A dimensionality of 30 was selected, which means each candidate solution in the population had 30 dimensions. The population size was set to 30 individuals, ensuring a diverse set of potential solutions to explore. The maximum number of iterations was set to 2000, allowing sufficient time for the algorithms to converge or terminate. Furthermore, to obtain robust results, each experiment was repeated 30 times (trail runs). The parameter settings specific to the developed M-DE algorithm were mentioned in a previous section, while the other comparative methods' parameter configurations can be found in their respective research papers. This ensures that the comparison is based on consistent settings for each algorithm.

In the presentation of the experimental results, boldface is used to highlight the best outcomes achieved by each algorithm. The results demonstrate the performance of the M-DE algorithm against the other methods, showcasing its effectiveness in solving both unimodal and multimodal optimization problems. To provide a comprehensive evaluation, the experimental results for the developed M-DE algorithm are presented alongside those of the other comparative methods. These results include the objective function values obtained for each algorithm on the benchmark functions. By comparing the performance of M-DE with the other methods, researchers can gauge the algorithm's efficiency and suitability for different problem types.

In conclusion, the experiments conducted on the standard benchmark suites demonstrate the efficiency and competitiveness of the developed M-DE algorithm. Its performance is thoroughly compared with other state-of-the-art methods, allowing researchers to gain insights into its strengths and weaknesses. The consistent parameter settings and rigorous evaluation process ensure a fair and meaningful comparison, providing valuable guidance for selecting the most appropriate optimization algorithm for specific problem domains.

4.1. Numerical analysis

The experimental results of developed M-DE on 6 unimodal and 7 multimodal standard benchmark functions equated with DE/rand/1[2], DE/best/1[39], DE/target-to-best/1[40], GDE[41] and PSODE[42]. Mean (mn) and standard deviation (std) over 30 trail runs of these methods reported in **Table 2**. The symbol "+" "≈" and "–" included in these tables respectively denote presentation of M-DE is better significantly, no substantial difference and inferior than equated methods. From this table, it can be noticed that - for whole unimodal and mulitimodal functions M-DE obtain the best optimal solutions; only for function F_6, F_{12} and F_{13} M-DE attain similar results with other algorithms. In between +/≈/–, M-DE secure maximum number of "+" i.e. 10, which shows M-DE can beat most of methods on all benchmark functions. Also, the less SD of M-DE on all benchmark functions shows its solution stability compared to others. Moreover, the statistical *t*-test[43] results on 6 unimodal and 7 multimodal benchmark suites presented in **Table 3**. It should be noticed that from this table, most of the p-values are below 0.05, which illustrate that convergence of M-DE is enhanced successfully. Additionally, the Friedman's ranking test[43] testified on all associated algorithms on 6 unimodal and 7 multimodal benchmark suites and results described in **Table 4**. It specifies that, projected M-DE reaches the best ranking in all functions.

Finally, to measure the significance of experimental results of M-DE with others nFEs (number of function evaluations) SR% (success rate %) considered on 6 unimodal and 7 multimodal benchmark functions and reported in **Table 5**. Where, $\text{success Rate} = \frac{\text{number of successful run}}{\text{total runs}}$ (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). This table shows that proposed M-DE has less number of average function evaluations and similar or highest percentage of the success rate on each benchmark function compared to others. It illustrates that M-DE has best convergence ability and higher reliability with others.

Table 2. Comparisons results on 6 unimodal and 7 multimodal benchmark functions.

Func	Measure	DE/rand/1[2]	DE/best/1[39]	DE/target-to-best/1[40]	GDE[41]	PSODE[42]	M-DE
F_1	mn	1.39×10^{-36}	1.61×10^{-39}	2.95×10^{-41}	4.82×10^{-46}	1.44×10^{-150}	0
	std	1.19×10^{-36}	1.38×10^{-39}	2.69×10^{-41}	1.13×10^{-45}	5.72×10^{-150}	0
F_2	mn	7.48×10^{-19}	4.28×10^{-20}	9.35×10^{-21}	2.87×10^{-21}	5.14×10^{-84}	0
	std	3.59×10^{-19}	3.64×10^{-20}	3.64×10^{-21}	4.99×10^{-21}	1.43×10^{-83}	0
F_3	mn	1.17×10^{-20}	1.09×10^{-22}	4.68×10^{-24}	1.33×10^{-24}	2.56×10^{-41}	0
	std	9.57×10^{-21}	1.34×10^{-22}	3.77×10^{-24}	2.77×10^{-24}	1.02×10^{-41}	0
F_4	mn	3.04×10^{-13}	2.69×10^{-14}	3.94×10^{-15}	1.04×10^{-14}	2.05×10^{-12}	0
	std	2.35×10^{-13}	2.81×10^{-14}	1.69×10^{-15}	2.91×10^{-14}	1.25×10^{-14}	0
F_5	mn	2.32×10^{-12}	2.28×10^{-21}	5.98×10^{-22}	1.32×10^{-15}	2.83×10^{-54}	0
	std	3.35×10^{-12}	3.29×10^{-21}	1.26×10^{-22}	3.37×10^{-15}	3.26×10^{-60}	0
F_6	mn	0	0	0	0	0	0
	std	0	0	0	0	0	0
F_7	mn	1.78×10^{-03}	2.01×10^{-03}	1.69×10^{-03}	1.32×10^{-03}	2.63×10^{-04}	2.84×10^{-12}
	std	6.76×10^{-04}	8.38×10^{-04}	7.76×10^{-04}	6.04×10^{-04}	1.98×10^{-04}	4.57×10^{-16}
F_8	mn	$-2.41\times10^{+02}$	$-6.61\times10^{+02}$	$-5.01\times10^{+02}$	$-2.78\times10^{+02}$	$-5.85\times10^{+04}$	$-12.5\times10^{+04}$
	std	$3.61\times10^{+02}$	$3.74\times10^{+02}$	$1.34\times10^{+02}$	$1.87\times10^{+02}$	$1.22\times10^{+01}$	1.02×10^{-02}
F_9	mn	$1.89\times10^{+01}$	$6.44\times10^{+00}$	$2.19\times10^{+01}$	$5.59\times10^{+00}$	5.79×10^{-15}	0
	std	$3.22\times10^{+00}$	$1.64\times10^{+00}$	$3.39\times10^{+00}$	$1.57\times10^{+00}$	1.21×10^{-14}	0
F_{10}	mn	4.44×10^{-15}	5.14×10^{-15}	4.44×10^{-15}	7.99×10^{-15}	1.19×10^{-14}	1.01×10^{-15}
	std	0	1.48×10^{-15}	0	2.90×10^{-15}	2.05×10^{-15}	2.51×10^{-16}
F_{11}	mn	1.81×10^{-02}	1.21×10^{-02}	3.12×10^{-02}	8.88×10^{-02}	1.58×10^{-02}	0
	std	9.15×10^{-02}	1.02×10^{-01}	8.61×10^{-02}	4.69×10^{-02}	2.38×10^{-02}	0
F_{12}	mn	4.71×10^{-32}	4.71×10^{-32}	4.71×10^{-32}	4.71×10^{-32}	4.71×10^{-32}	4.71×10^{-32}
	std	1.14×10^{-47}	1.15×10^{-47}	1.15×10^{-47}	1.15×10^{-47}	2.58×10^{-41}	1.25×10^{-48}
F_{13}	mn	1.34×10^{-32}	1.34×10^{-32}	1.34×10^{-32}	1.34×10^{-32}	1.34×10^{-32}	1.34×10^{-32}
	std	2.89×10^{-48}	2.89×10^{-48}	2.88×10^{-48}	2.88×10^{-48}	5.68×10^{-41}	1.28×10^{-48}
No. of +/≈/−		0/3/10	0/3/10	0/3/10	0/3/10	0/3/10	

Table 3. The statistical t-test (p-values) for M-DE vs other algorithms.

Fun	DE/rand/1 vs M-DE	DE/best/1 vs M-DE	DE/target-to-best/1 vs M-DE	GDE vs M-DE	PSODE vs M-DE
F_1	4.012×10^{-16}	2.101×10^{-08}	2.102×10^{-18}	2.010×10^{-08}	1.104×10^{-10}
F_2	2.120×10^{-15}	1.011×10^{-10}	1.010×10^{-08}	2.002×10^{-10}	2.001×10^{-12}
F_3	1.011×10^{-10}	1.005×10^{-08}	2.001×10^{-10}	1.110×10^{-10}	1.004×10^{-08}
F_4	2.110×10^{-18}	2.014×10^{-04}	4.001×10^{-18}	1.100×10^{-12}	2.121×10^{-12}
F_5	$5.014e\times10^{-12}$	1.507×10^{-40}	2.801×10^{-21}	4.180×10^{-20}	1.004×10^{-18}
F_6	2.101×10^{-14}	1.112×10^{-18}	1.021×10^{-20}	2.102×10^{-16}	1.000×10^{-16}
F_7	1.010×10^{-10}	1.102×10^{-08}	1.001×10^{-10}	2.110×10^{-10}	1.001×10^{-08}
F_8	1.010×10^{-12}	4.001×10^{-18}	2.111×10^{-40}	2.021×10^{-40}	1.017×10^{-20}
F_9	2.104×10^{-19}	1.107×10^{-08}	2.001×10^{-10}	4.012×10^{-10}	2.218×10^{-08}
F_{10}	5.001×10^{-14}	4.044×10^{-10}	2.021×10^{-12}	1.010×10^{-10}	1.075×10^{-16}
F_{11}	4.101×10^{-18}	5.120×10^{-08}	5.015×10^{-08}	1.201×10^{-08}	1.001×10^{-08}
F_{12}	2.100×10^{-14}	1.102×10^{-10}	2.001×10^{-11}	1.005×10^{-10}	4.021×10^{-12}
F_{13}	1.005×10^{-18}	2.011×10^{-10}	5.008×10^{-10}	2.001×10^{-10}	4.240×10^{-08}

Table 4. Friedman's average ranking test of different algorithms.

Algorithms	Ranking
DE/rand/1	5.61
DE/best/1	5.03
DE/target-to-best/1	4.13
GDE	3.09
PSODE	2.15
M-DE	1.80

Table 5. Function evaluations and success rate % of different algorithms.

Func	Measure	DE/rand/1[2]	DE/best/1[39]	DE/target-to-best/1[40]	GDE[41]	PSODE[42]	M-DE
F_1	nFEs	118,197	112,408	91,496	72,081	18,204	**8519**
	SR	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
F_2	nFEs	115,441	109,849	91,354	66,525	15,067	**7867**
	SR	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
F_3	nFEs	110,254	125,474	87,014	75,408	18,564	**8655**
	SR	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
F_4	nFEs	145,874	128,555	105,547	95,874	20,500	**12,000**
	SR	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
F_5	nFEs	125,000	118,000	85,000	75,550	20,850	**18,500**
	SR	95.6%	96.02%	**100%**	**100%**	**100%**	**100%**
F_6	nFEs	130,500	125,800	95,600	65,000	35,000	**25,000**
	SR	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
F_7	nFEs	102,259	103,643	87,518	74,815	16,115	**10,182**
	SR	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
F_8	nFEs	125,500	118,500	116,700	95,000	65,000	**45,500**
	SR	85.02%	96.05%	98.11%	**100%**	**100%**	**100%**
F_9	nFEs	99,074	98,742	127,423	53,416	7701	**5627**
	SR	96.70%	**100%**	**100%**	**100%**	**100%**	**100%**
F_{10}	nFEs	125,543	118,926	100,000	76,646	29,757	**17,551**
	SR	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
F_{11}	nFEs	125,777	117,946	97,213	81,422	18,394	**9014**
	SR	60.00%	46.70%	56.70%	**100%**	**100%**	**100%**
F_{12}	nFEs	125,500	125,550	100,500	82,500	38,500	**25,000**
	SR	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
F_{13}	nFEs	120,500	120,200	110,500	85,000	44,500	**28,500**
	SR	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**

4.2. Graphical analysis

In this section, the convergence results of the developed Modified Differential Evolution (M-DE) algorithm are analyzed using convergence curves. Convergence curves are graphical representations that illustrate the algorithm's speed and accuracy in approaching the optimal solution compared to other methods. The convergence curves of M-DE are plotted alongside those of other state-of-the-art optimization algorithms on eight standard benchmark suites (F_1, F_5, F_6, F_7, F_8, F_9, F_{10}, and F_{11}), providing valuable insights into the algorithm's performance.

Figures 2a–h displays the convergence curves for each benchmark suite, where the number of iterations is shown on the x-axis and the objective function values obtained from each algorithm on the same population or seed are depicted on the y-axis. The objective function values represent the quality of the candidate solutions generated by the algorithms, and lower values indicate better performance in terms of convergence accuracy. Upon analyzing the convergence curves, several key observations can be made:

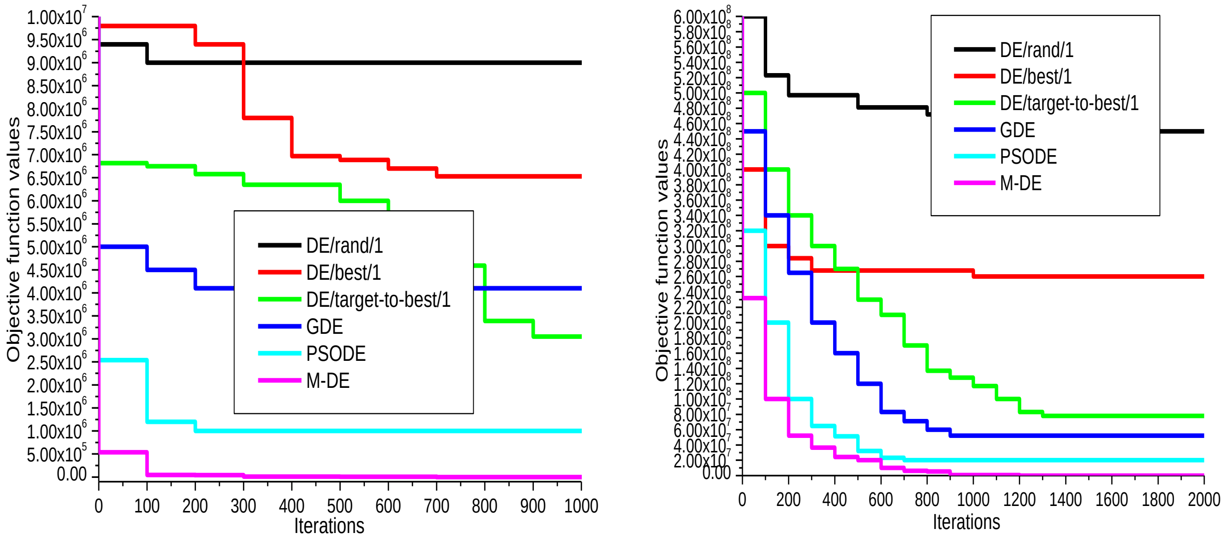
Quicker Convergence: The convergence curves for M-DE generally exhibit faster convergence compared to other methods. The algorithm efficiently refines its solutions over iterations, approaching the optimal solution rapidly. This is evident from the steeper slope of the convergence curves for M-DE, indicating a faster reduction in objective function values with increasing iterations.

Better Accuracy: In most cases, the convergence curves for M-DE show better accuracy in reaching the optimal solution compared to other methods. The objective function values achieved by M-DE are consistently lower, indicating superior solutions in terms of optimization performance.

Effective Escape from Local Optima: The observed faster convergence and better accuracy of M-DE suggest its capability to effectively escape local optima. Local optima are suboptimal solutions that are prevalent in complex optimization landscapes. The ability to jump out of local optima is a crucial aspect of any optimization algorithm, as it allows the algorithm to explore diverse regions of the search space and find global optima.

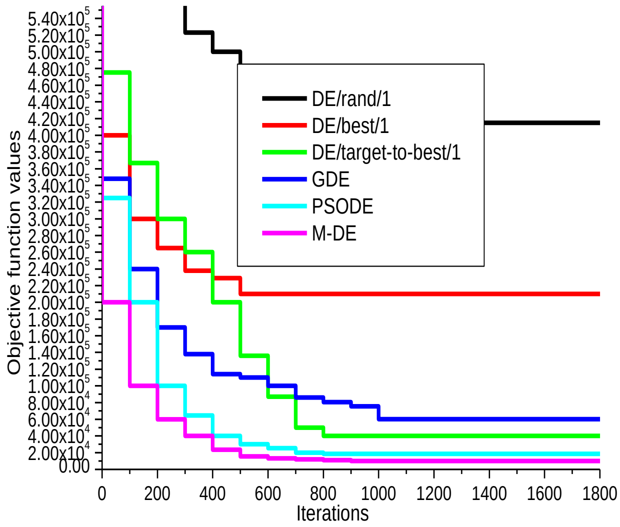
The convergence curves provide compelling evidence of the M-DE algorithm's superiority over other methods in terms of convergence speed and accuracy. Its effectiveness in escaping local optima suggests that the novel mutation strategy and time-varying control parameters incorporated into M-DE have successfully enhanced its exploration capability.

The superior performance of M-DE observed in these convergence curves reinforces its potential as a robust and efficient optimization algorithm for a wide range of problem domains. It offers researchers and practitioners a reliable tool for tackling complex optimization challenges, especially when dealing with multimodal or deceptive landscapes where local optima are prevalent. The convergence curves presented in **Figures 2a–h** demonstrate the rapid convergence and high accuracy of the developed M-DE algorithm compared to other state-of-the-art optimization methods. Its ability to escape local optima effectively highlights the efficacy of the novel mutation strategy and time-varying control parameters. These results underscore M-DE's potential as a powerful optimization tool, providing valuable insights for researchers and practitioners seeking efficient and effective solutions to various real-world optimization problems.

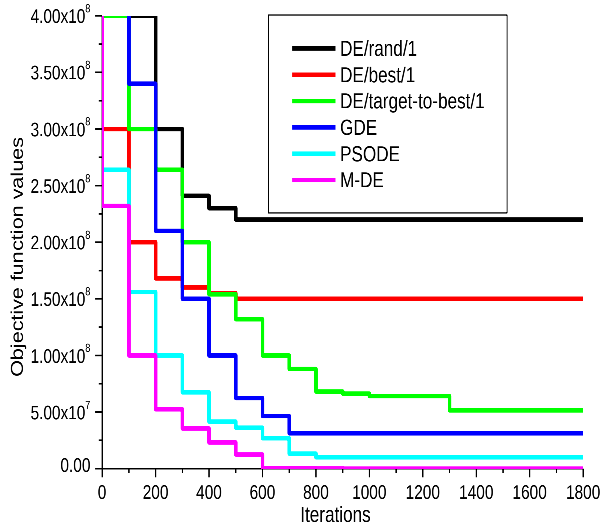


(a). F_1 **(b)**. F_5

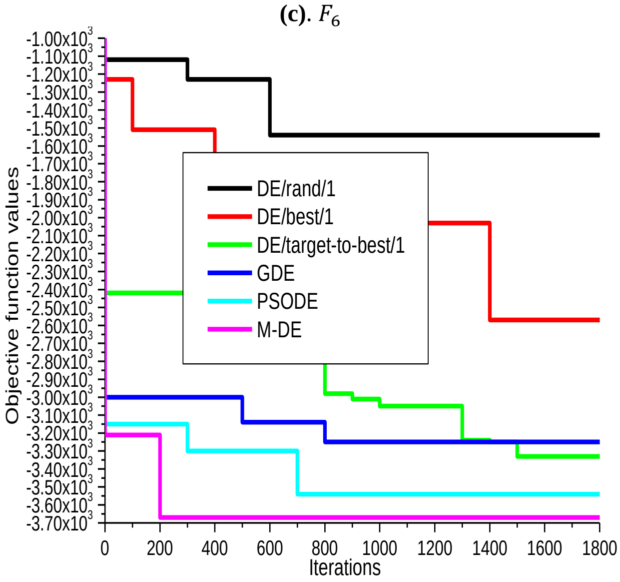
Figure 2. (*Continued*).



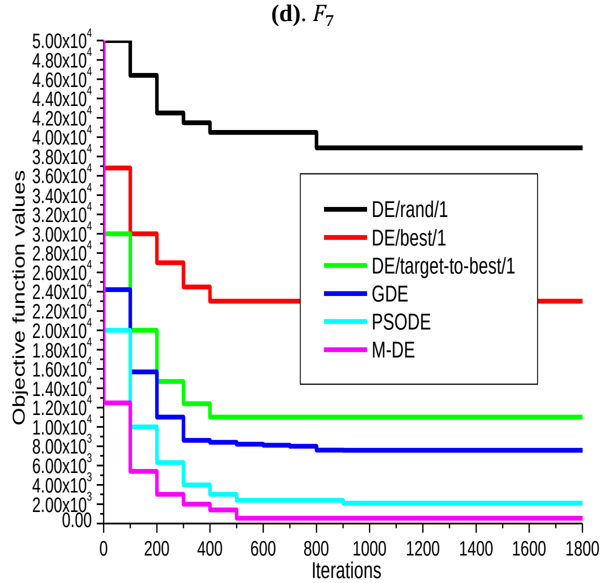
(c). F_6



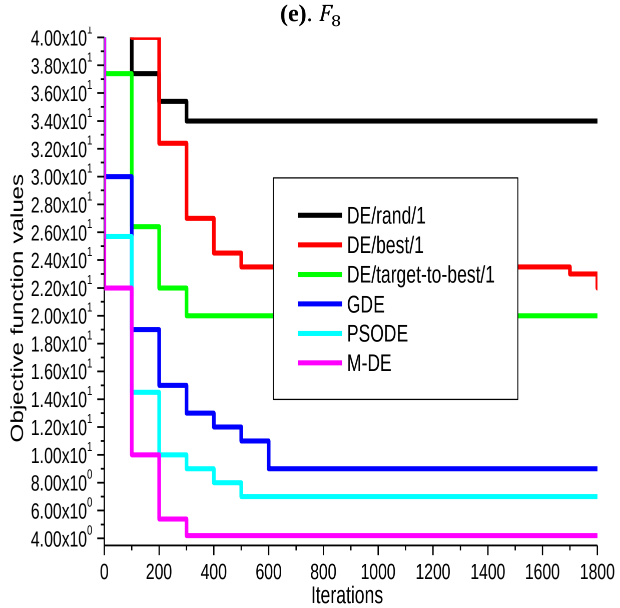
(d). F_7



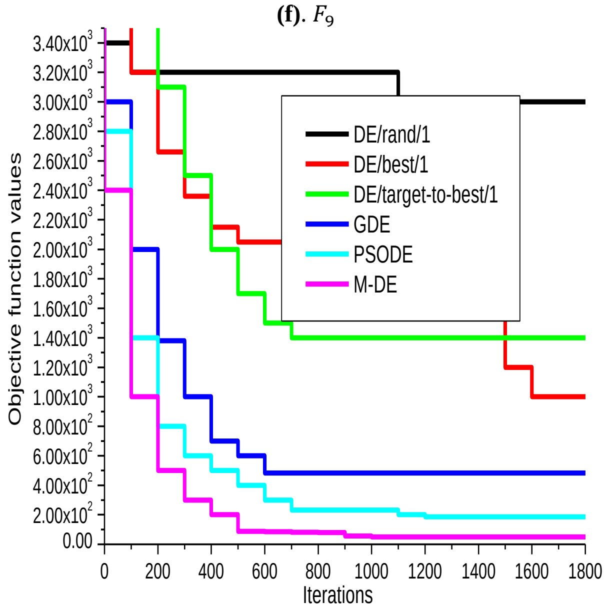
(e). F_8



(f). F_9



(g). F_{10}



(h). F_{11}

Figure 2. Convergence curves of M-DE with other methods.

In this section, the performance of the developed Modified Differential Evolution (M-DE) algorithm is further analyzed and compared to other state-of-the-art optimization methods using computational time, empirical distribution of normalized success performance, and performance chart.

Figure 3 displays the computational time (in seconds) of M-DE alongside other algorithms on the six unimodal and seven multimodal benchmark functions. The graph demonstrates that M-DE consistently provides better results with significantly less computation time. This indicates M-DE's powerful search performance, as it achieves superior optimization outcomes within shorter timeframes. The reduced computational time is of utmost importance in real-world applications where efficiency is critical for solving complex optimization problems.

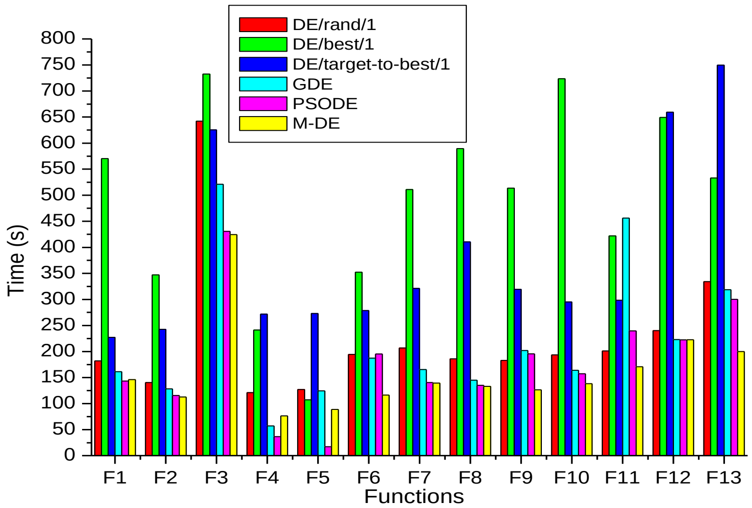


Figure 3. Shows computational time (in seconds) required by the Modified Differential Evolution (M-DE) algorithm.

Next, the empirical distribution of normalized success performance, a commonly used metric for performance comparison, is used to assess the overall performance of M-DE against the other algorithms. **Figure 4** illustrates the empirical distribution of normalized success performance[44], showing how frequently each method outperforms the others across multiple trials. From the figure, it is evident that M-DE consistently outperforms the compared methods, achieving higher success rates and demonstrating its robustness and reliability in finding high-quality solutions.

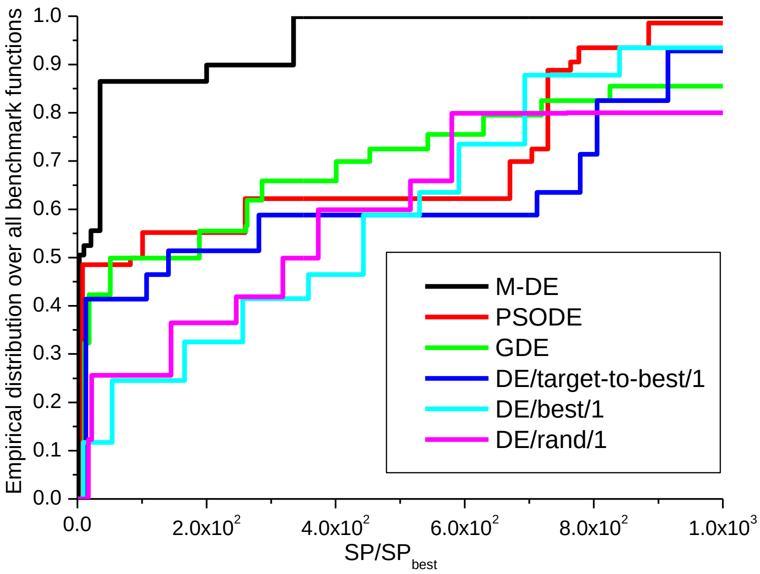


Figure 4. Illustrates the empirical distribution of normalized success performance, which compares the effectiveness of the Modified Differential Evolution (M-DE) algorithm with other state-of-the-art optimization methods**.**

Figure 5 presents the performance chart[45] that further validates the superiority of M-DE over the other algorithms. The performance chart provides a comprehensive comparison of each algorithm's performance on multiple benchmark functions, enabling a holistic assessment of their capabilities. In this chart, M-DE's performance is shown to be the greatest among all the compared methods, indicating its effectiveness in solving a wide range of unconstrained optimization problems. The cumulative evidence from **Figure 3**, **Figure 4**, and **Figure 5** reinforces the conclusion that the projected M-DE algorithm is a powerful optimizer for unconstrained optimization problems. Its superior performance in terms of computational efficiency, success rate, and overall performance compared to state-of-the-art methods underscores its effectiveness as a reliable and efficient optimization tool.

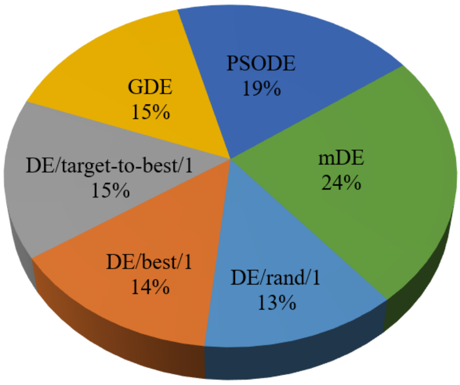


Figure 5. The performance of the Modified Differential Evolution (M-DE) algorithm was compared with that of other optimization methods.

The robustness and versatility of M-DE make it well-suited for various real-world optimization tasks, ranging from engineering and logistics to finance and data science. The novel mutation strategy and time-varying control parameters incorporated into M-DE have proven to be instrumental in enhancing its exploration and exploitation capabilities, enabling it to find high-quality solutions more efficiently than its counterparts.

The thorough analysis of M-DE's performance through computational time, empirical distribution of normalized success performance, and performance chart confirms its status as a superior optimization algorithm. Its ability to consistently outperform other methods in terms of efficiency and solution quality solidifies its position as a powerful optimizer for unconstrained optimization problems. Researchers and practitioners can confidently rely on M-DE to tackle challenging optimization tasks and achieve optimal solutions in various real-world applications.

5. Conclusion and future works

In this research article, we proposed a modified differential evolution (M-DE) algorithm for tackling unimodal and multimodal problem optimization. The M-DE algorithm incorporates a novel mutation scheme inspired by particle swarm optimization, which effectively balances the diversity of the population. Additionally, we introduced time-varying mutant control parameters that aid individuals in escaping local optima, thereby enhancing the algorithm's exploration and exploitation abilities. The integration of these features into M-DE has resulted in significant improvements in convergence speed, making it a more efficient optimization approach.

To evaluate the performance of M-DE, extensive experiments were conducted on 6 unimodal and 7 multimodal benchmark suites. The results of the experiments were compared against several state-of-the-art differential evolution variants and particle swarm optimization. The experimental findings clearly demonstrate that M-DE outperforms the compared algorithms in terms of both solution quality and computational efficiency. M-DE consistently provided better results across all benchmark suites, achieving higher success rates and requiring less time to converge.

6. Future scope

The proposed M-DE algorithm shows great promise as an effective variant of differential evolution for solving unimodal and multimodal optimization problems. As such, there are several avenues for future research and potential extensions of this work:

1) Hybrid Approaches: Investigate the potential benefits of combining M-DE with other optimization techniques, such as genetic algorithms or simulated annealing, to create hybrid algorithms that leverage the strengths of each approach.
2) Parameter Tuning: Conduct more in-depth studies on fine-tuning the control parameters of M-DE to further enhance its performance on specific problem classes and real-world applications.
3) Dynamic Environments: Explore the adaptability of M-DE in dynamic optimization scenarios, where the underlying optimization landscape changes over time.
4) Constraint Handling: Extend M-DE to handle constraint optimization problems, allowing it to efficiently handle optimization tasks with constraints.
5) Real-World Applications: Apply M-DE to various real-world optimization problems in fields like engineering, finance, logistics, and other domains to assess its practical effectiveness.
6) Theoretical Analysis: Undertake theoretical analyses to gain a deeper understanding of the dynamics and convergence properties of M-DE in different problem scenarios.
7) Parallelization: Investigate methods to parallelize M-DE for high-performance computing environments, enabling faster convergence on large-scale optimization tasks.

By addressing these future research directions, M-DE can be further refined and adapted for various complex optimization challenges, solidifying its position as a valuable tool for solving real-world optimization problems efficiently.

Author contributions

Conceptualization, PT; methodology, PT, VNM and RPP; software, PT and RPP; validation, PT; formal analysis, PT; investigations, PT; resources, VNM; data curation, PT; writing—original draft preparation, PT; writing—review and editing, PT; visualization, RPP; supervision, VNM; project administration, VNM, funding acquisition, RPP. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Parouha RP, Das KN. Parallel hybridization of differential evolution and particle swarm optimization for constrained optimization with its application. International Journal of System Assurance Engineering and Management 2016; 7(S1): 143–162. doi: 10.1007/s13198-015-0354-6
2. Storn R, Price K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 1997; 11(2): 341–359. doi: 10.1023/A:1008202821328
3. Dhanalakshmy DM, Jeyakumar G, Velayutham CS. Empirical investigations on evolution strategies to self-adapt the mutation and crossover parameters of differential evolution algorithm. International Journal of Intelligent Systems Technologies and Applications 2021; 20(2): 103–125. doi: 10.1504/ijista.2021.119028
4. Wang Y, Li B, Weise T. Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems. Information Sciences 2010; 180(12): 2405–2420. doi: 10.1016/j.ins.2010.02.015

5. Dragoi EN, Curteanu S, Galaction AI, Cascaval D. Optimization methodology based on neural networks and self-adaptive differential evolution algorithm applied to an aerobic fermentation process. Applied Soft Computing 2013; 13(1): 222–238. doi: 10.1016/j.asoc.2012.08.004
6. Mesejo P, Ugolotti R, Di Cunto F, et al. Automatic hippocampus localization in histological images using differential evolution-based deformable models. Pattern Recognition Letters 2013; 34(3): 299–307. doi: 10.1016/j.patrec.2012.10.012
7. Li X, Hu C, Yan X. Chaotic differential evolution algorithm based on competitive coevolution and its application to dynamic optimization of chemical processes. Intelligent Automation & Soft Computing 2013; 19(1): 85–98. doi: 10.1080/10798587.2013.771437
8. Das KN, Parouha RP. Optimization with a novel hybrid algorithm and applications. OPSEARCH 2016; 53(3): 443–473. doi: 10.1007/s12597-015-0240-7
9. Parouha RP, Das KN. Economic load dispatch using memory based differential evolution. International Journal of Bio-Inspired Computation 2018; 11(3): 159–170. doi: 10.1504/ijbic.2018.091700
10. Neri F, Tirronen V. Recent advances in differential evolution: A survey and experimental analysis. Artificial Intelligence Review 2010; 33(1–2): 61–106. doi: 10.1007/s10462-009-9137-2
11. Das S, Suganthan PN. Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation 2011; 15(1): 4–31. doi: 10.1109/tevc.2010.2059031
12. Das S, Mullick SS, Suganthan PN. Recent advances in differential evolution—An updated survey. Swarm and Evolutionary Computation 2016; 27: 1–30. doi: 10.1016/j.swevo.2016.01.004
13. Eltaei T, Mahmood A. Differential evolution: A survey and analysis. Applied Sciences 2018; 8(10): 1945. doi: 10.3390/app8101945
14. Opara KR, Arabas J. Differential evolution: A survey of theoretical analyses. Swarm and Evolutionary Computation 2019; 44: 546–558. doi: 10.1016/j.swevo.2018.06.010
15. Bilal, Pant M, Zaheer H, et al. Differential evolution: A review of more than two decades of research. Engineering Applications of Artificial Intelligence 2020; 90: 103479. doi: 10.1016/j.engappai.2020.103479
16. 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. doi: 10.1016/j.aej.2021.09.013
17. Lampinen J, Zelinka I. On stagnation of the differential evolution algorithm. In: Proceedings of MENDEL 2000, 6th International Mendel Conference on Soft Computing; 7–9 June 2000; Brno, Czech Republic. pp. 76–83.
18. Coelho L dos S, Souza RCT, Mariani VC. Improved differential evolution approach based on cultural algorithm and diversity measure applied to solve economic load dispatch problems. Mathematics and Computers in Simulation 2009; 79(10): 3136–3147. doi: 10.1016/j.matcom.2009.03.005
19. Mallipeddi R, Suganthan PN, Pan QK, Tasgetiren MF. Differential evolution algorithm with ensemble of parameters and mutation strategies. Applied Soft Computing 2011; 11(2): 1679–1696. doi: 10.1016/j.asoc.2010.04.024
20. Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters. IEEE Transactions on Evolutionary Computation 2011; 15(1): 55–66. doi: 10.1109/tevc.2010.2087271
21. Gong W, Fialho A, Cai Z, Li H. Adaptive strategy selection in differential evolution for numerical optimization: An empirical study. Information Sciences 2011; 181(24): 5364–5386. doi: 10.1016/j.ins.2011.07.049
22. Gong W, Zhou A, Cai Z. A multioperator search strategy based on cheap surrogate models for evolutionary optimization. IEEE Transactions on Evolutionary Computation 2015; 19(5): 746–758. doi: 10.1109/tevc.2015.2449293
23. Wu G, Mallipeddi R, Suganthan PN, et al. Differential evolution with multipopulation based ensemble of mutation strategies. Information Sciences 2016; 329: 329–345. doi: 10.1016/j.ins.2015.09.009
24. Zou D, Li S, Wang GG, et al. An improved differential evolution algorithm for the economic load dispatch problems with or without valve-point effects. Applied Energy 2016; 181: 375–390. doi: 10.1016/j.apenergy.2016.08.067
25. Neto JXV, Reynoso-Meza G, Ruppel TH, et al. Solving non-smooth economic dispatch by a new combination of continuous GRASP algorithm and differential evolution. International Journal of Electrical Power & Energy Systems 2017; 84: 13–24. doi: 10.1016/j.ijepes.2016.04.012
26. Zhang Q, Zou D, Duan N, Shen X. An adaptive differential evolutionary algorithm incorporating multiple mutation strategies for the economic load dispatch problem. Applied Soft Computing 2019; 78: 641–669. doi: 10.1016/j.asoc.2019.03.019
27. Ma H, Shen S, Yu M, et al. Multi-population techniques in natureinspired optimization algorithms: A comprehensive survey. Swarm and Evolutionary Computation 2019; 44: 365–387. doi: 10.1016/j.swevo.2018.04.011
28. Noman N, Iba H. Accelerating differential evolution using an adaptive local search. IEEE Transactions on Evolutionary Computation 2008; 12(1): 107–125. doi: 10.1109/tevc.2007.895272
29. Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of IEEE İnternational Conference on Neural Networks 1995; 4: 1942–1948.
30. Jia D, Zheng G, Khan MK. An effective memetic differential evolution algorithm based on chaotic local search. Information Sciences 2011; 181(15): 3175–3187. doi: 10.1016/j.ins.2011.03.018

31. Dhaliwal JS, Dhillon J. Profit based unit commitment using memetic binary differential evolution algorithm. Applied Soft Computing 2019; 81: 105502. doi: 10.1016/j.asoc.2019.105502
32. Zuo M, Dai G, Peng L, et al. A case learning-based differential evolution algorithm for global optimization of interplanetary trajectory design. Applied Soft Computing 20220; 94: 106451. doi: 10.1016/j.asoc.2020.106451
33. 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. doi: 10.1007/s11831-021-09532-7
34. Parouha RP, Das KN. A memory based differential evolution algorithm for unconstrained optimization. Applied Soft Computing 2016; 38: 501–517. doi: 10.1016/j.asoc.2015.10.022
35. Brest J, Greiner S, Boskovic B, et al. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Transactions on Evolutionary Computation 2006; 10(6): 646–657. doi: 10.1109/tevc.2006.872133
36. Qin AK, Huang VL, Suganthan PN. Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Transactions on Evolutionary Computation 2009; 13(2): 398–417. doi: 10.1109/tevc.2008.927706
37. Das KN, Parouha RP. An ideal tri-population approach for unconstrained optimization and applications. Applied Mathematics and Computation 2015; 256: 666–701. doi: 10.1016/j.amc.2015.01.076
38. Verma P, Parouha RP. An advanced hybrid algorithm for engineering design optimization. Neural Processing Letters 2021; 53(5): 3693–3733. doi: 10.1007/s11063-021-10541-7
39. 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. doi: 10.1109/tevc.2008.2009457
40. Cheshmehgaz HR, Desa MI, Wibowo A. Effective local evolutionary searches distributed on an island model solving bi-objective optimization problems. Applied Intelligence 2013; 38(3): 331–356. doi: 10.1007/s10489-012-0375-7
41. Han MF, Liao SH, Chang JY, Lin CT. Dynamic group-based differ-ential evolution using a self-adaptive strategy for global optimization problems. Applied Intelligence 2013; 39(1): 41–56. doi: 10.1007/s10489-012-0393-5
42. 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): 629–664. doi: 10.1016/j.asoc.2009.08.031
43. Derrac J, Garcia 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. doi: 10.1016/j.swevo.2011.02.002
44. García S, Molina D, Lozano M, Herrera F. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization. Journal of Heuristics 2009; 15: 617–644. doi: 10.1007/s10732-008-9080-4
45. Parouha RP, Das KN. DPD: An intelligent parallel hybrid algorithm for economic load dispatch problems with various practical constraints. Expert Systems with Applications 2016; 63: 295–309. doi: 10.1016/j.eswa.2016.07.012