UDE-IV: An Enhanced Unified Differential Evolution Algorithm for CEC 2025 Constrained Optimization Problems

Dikshit Chauhan and Anupam Trivedi

***Abstract*—In this paper, an enhanced unified differential evolution algorithm, named UDE-IV, is presented for real parameter-constrained optimization problems (COPs). UDE-IV builds upon its predecessors, UDE-II and UDE-III, which attained first rank in the CEC 2018 and CEC 2024 competitions, respectively. To design UDE-IV, we extensively analyzed and addressed the limitations of UDE-III. UDE-IV employs four trial vector generation strategies: DE/rand/1, DE/current-to-rand/1, DE/current-to-$p$best/1, and DE/rand-to-$p$best/1. The algorithm features a dual population approach, dividing the current population into two sub-populations at each generation. The top sub-population applies DE/rand/1, DE/current-to-rand/1, and DE/current-to-$p$best/1 to each target vector. In contrast, the bottom sub-population employs a strategy adaptation mechanism and utilizes a single strategy per target vector. This mechanism dynamically adjusts the probabilities of the three trial vector generation strategies based on their past success in generating promising solutions. To further boost the exploration capabilities compared to UDE-III, UDE-IV probabilistically uses DE/current-to-$p$best/1 and DE/rand-to-$p$best/1 within the bottom sub-population. UDE-IV retains the parameter adaptation method from UDE-III, which is based on the LSHADE44 approach. It also adopts the same constraint handling mechanism as in UDE-III which is based on a combination of the feasibility rule and the epsilon-constraint handling technique, as proposed in C^{2}oDE. The novel stagnation avoidance strategy introduced in UDE-III is also preserved, however, its parameters are optimally tuned in UDE-IV for improved performance. Furthermore, ranking-based parent selection is removed in UDE-IV as it was found to hinder the algorithm's efficacy. The proposed algorithm UDE-IV is rigorously tested on the 28 benchmark 30D problems from the CEC 2025 competition on constrained real parameter optimization. Comparative experimental results with UDE-III demonstrate UDE-IV's superior efficacy in solving complex real-parameter COPs.**

***Keywords*—Constrained optimization, differential evolution, parameter adaptation, stagnation, strategy adaptation.**

I. Introduction

Differential evolution (DE) is a simple yet powerful evolutionary algorithm EA proposed by Price and Storn [1] in 1995 for real parameter optimization. DE differs from traditional EAs in that it perturbs the current generation population members with the scaled differences of randomly selected and distinct population members. Over the last two decades, DE has been applied to a plethora of real-world problems and classical benchmark problems, owing to its simplicity and robust nature. Interested readers are referred to [2], [3], [4], [5] for a comprehensive review of DE.

In this paper, we present a DE algorithm named Unified Differential Evolution IV (UDE-IV) to solve constrained real-parameter optimization problems. The roots of UDE IV lie in UDE-II [6] and UDE-III [7], which were the winners of the CEC 2018 and CEC 2024 constrained real parameter optimization competition. UDE-II, UDE-III, and UDE-IV share the following principles. They are all built on the philosophy of unified algorithm design, comprehensive offline + online learning, and black-box thinking.

Unified algorithm design implies that to design a powerful algorithm, it is important to pay attention to all the design elements of the algorithm, not just one or a few. Design elements are the components (i.e., operators) of an algorithm or its parameters on which its performance is critically dependent. In other words, unified algorithm design means looking at the algorithm as a whole instead of strengthening just one or a few design elements. UDE-II, for example, took into consideration the following design elements: 1) DE trial vector generation strategy/strategy pool, 2) strategy adaptation, 3) control parameter adaptation, 4) parent selection strategy, and 5) constraint handling technique. In UDE-III, we further focused on the holistic design of the algorithm and considered additional important design elements. One of the design elements is the stagnation strategy. In UDE-III, we integrated a novel stagnation strategy that helps overcome stagnation by utilizing an archive based on unsuccessful trial vectors, which are generally discarded in DE.

Offline + online learning consists of two parts - offline and online learning. In our design philosophy, offline learning indicates that it is important for the algorithm designer to learn offline, i.e., from other relevant high-performing algorithms proposed in the literature. On the other hand, online learning means that the algorithm learns or improves in an online manner from the data that is generated during the evolution. In UDE-II, offline learning was implemented in the sense that its design is inspired by many state-of-the-art algorithms. UDE-II's design leverages many existing DE algorithms: JADE's DE trial vector generation strategy [8], CoDE's strategy pool [9], SaDE's strategy adaptation [10], SHADE's parameter adaptation [11], rank-based DE's parent selection strategy [12], and C^{2}oDE's constraint handling technique [13]. In UDE-III, we extended offline learning, and to design the stagnation

Anupam Trivedi is with the Department of Electrical and Computer Engineering of the National University of Singapore, Singapore, *email:* eleatr@nus.edu.sg.

Dikshit Chauhan is with the Department of Electrical and Computer Engineering of the National University of Singapore, Singapore, *e-mail:* diksh_06@nus.edu.sg

strategy, we took inspiration from the best-discarded vector selection (BDVS) strategy [14].

Online learning refers to an algorithm's ability to improve by using the data that is generated during the optimization process. UDE-II achieved this by incorporating online strategy adaptation and parameter adaptation mechanisms. In UDE-III, we extended online learning and leveraged the large number of unsuccessful trial vectors that were generated during the evolution. Specifically, we incorporated an external archive to store the unsuccessful trial vectors that are generated during the evolution. The archive size is always maintained to be less than equal to the population size by using the superiority of feasibility principle. In the stagnation strategy, when a stagnated individual meets the conditions to be replaced, an individual is randomly selected from the external archive to replace the stagnated individual. This way, a large number of unsuccessful trial vectors that are generally discarded during the evolution are utilized to improve the performance of the algorithm online.

The black box thinking term is adopted from the famous book by Matthew Syed titled "Black Box Thinking: Marginal Gains and the Secrets of High Performance" [15]. In this book, Matthew Syed argues that in order to succeed, it is of utmost importance to analyze the failures and address them. In our algorithm design philosophy, black box thinking implies that in order to improve the performance of an algorithm, it is essential to determine its weaknesses and exhaustively target them. Thus, UDE-III has been designed by comprehensively analyzing the weaknesses of UDE-II and addressing them, while UDE-IV has been designed by investigating and overcoming the shortcomings of UDE-III.

Overall, the proposed UDE-IV is a significantly enhanced version of UDE-III. Like its predecessors, UDE-II and UDE-III, UDE-IV, too, is a dual-population algorithm. UDE-III uses three trial vector generation strategies - DE/rand/1, DE/current-to-rand/1, and DE/current-to-$p$best/1, whereas UDE-IV uses four trial vector generation strategies - DE/rand/1, DE/current-to-rand/1, DE/current-to-$p$best/1, and DE/rand-to-$p$best/1. At each generation, UDE-IV divides the current population into two sub-populations. In the top sub-population, it employs three trial vector generation strategies (DE/rand/1, DE/current-to-rand/1, and DE/current-to-$p$best/1) on each target vector. The bottom sub-population in UDE-IV applies only one trial vector generation strategy on each target vector. Specifically, the bottom sub-population employs strategy adaptation, wherein the probability of the three trial vector generation strategies implemented in the top sub-population is adapted in every generation by learning from their experiences in generating successful/unsuccessful trial vectors across the whole population in the previous learning period. To enhance the exploration capabilities, UDE-IV probabilistically uses DE/current-to-$p$best/1 and DE/rand-to-$p$best/1 in the bottom sub-population. In UDE-IV, the ranking-based mutation used in UDE-III is removed as it was found to lead to premature convergence on several problems. The parameter adaptation, constraint handling technique, and stagnation strategy, as used in UDE-III, are retained in UDE-IV. However, the parameters related to the stagnation strategy are optimally tuned in UDE-IV. In summary, as compared to UDE-III, UDE-IV differs in terms of strategy pool, which is designed to enhance the exploration capabilities, and exclusion of ranking-based mutation, which is done to prevent premature convergence and tuning of stagnation strategy parameters. The proposed UDE-IV algorithm is tested on the 28 benchmark 30D problems provided for the CEC 2025 competition on constrained real parameter optimization [16]. The experimental results demonstrate that UDE-IV is significantly better than its predecessor, UDE-III, in solving constrained real parameter optimization problems.

The rest of the paper is organized as follows. The basics of DE are described in Section II. The weaknesses of UDE-III and the scope of potential improvements are briefly discussed in Section III. In Section IV, the proposed algorithm UDE-IV is presented. The modifications made in UDE-III to design UDE-IV are summarized in Section V. The parameter settings and experimental results are summarized in Section VI. Finally, the paper is concluded in Section VII.

II. Differential Evolution

A. *Differential Evolution*

Differential evolution consists of three basic steps: mutation, crossover, and selection. The basics of DE are presented below.

B. *Initialization*

The initial population is generated randomly. Suppose the k-th individual of the population at generation G is denoted by $x_{k,G}$ (where $x_{k,G} = [x_{1,k,G}, x_{2,k,G}, ..., x_{D,k,G}]$, D being the number of dimensions or decision variables). The j-th decision variable of the k-th individual is randomly initialized for the initial population (at $G = 1$) as

$$x_{j,k,1} = x_{j,min} + rand_{k,j}[0,1].(x_{j,max} - x_{j,min}) \quad (1)$$

where $x_{j,min}$ and $x_{j,max}$ are the minimum and maximum bounds of the j-th decision variable, respectively and $rand_{k,j}[0,1]$ is an uniformly distributed random number lying between 0 and 1

C. *Mutation*

At generation $G+1$, corresponding to k-th individual in the population, $x_{k,G}$ (called target vector in DE literature), DE creates a mutant individual $v_{k,G+1}$ (where $v_{k,G+1} = [v_{1,k,G+1}, v_{2,k,G+1}, ..., v_{D,k,G+1}]$, D being the number of decision variables) through mutation. There are several DE variants in the literature and they differ mainly in the way mutation operation is executed. DE/rand/1 is one of the most popular classical DE variants and the equation for mutation operation in DE/rand/1 is as follows:

$$DE/rand/1: \; v_{k,G+1} = x_{r_1^k,G} + F(x_{r_2^k,G} - x_{r_3^k,G}) \quad (2)$$

where r_1^k, r_2^k and r_3^k are mutually exclusive and randomly chosen indices from $[1, N_p]$ and are also different from the base index k (where N_p is the population size). The scaling factor F is a control parameter and lies in the range $[0, 2]$. A smaller value of F promotes exploitation while a larger value of F promotes exploration

D. Crossover

After generating the mutant individual $v_{k,G+1}$ through mutation, the crossover comes into operation. The two most popular crossover methods in DE literature are the binomial crossover and the exponential crossover. Here, we describe the binomial crossover. In the binomial crossover, the mutant individual $v_{k,G+1}$ exchanges its components with the target individual $x_{k,G}$ with a probability $CR \in [0,1]$ to form the trial individual $u_{k,G+1}$ (where $u_{k,G+1} = [u_{1,k,G+1}, u_{2,k,G+1}, ..., u_{D,k,G+1}]$, D being the number of decision variables), according to the following condition:

$$u_{j,k,G+1} = \begin{cases} v_{j,k,G+1} & if\ (rand_{k,j}[0,1] \leq CR \text{ or } j = j_{rand}) \\ x_{j,k,G} & otherwise \end{cases} \tag{3}$$

where $rand_{k,j}[0,1]$ is a uniformly distributed random number and $j_{rand} \in [1,2,...,D]$ is a randomly chosen index that ensures that the trial individual gets at least one component from the mutant individual.

E. Selection

In the selection or the replacement step, a one-to-one comparison is performed between the target vector and the trial vector, and the fitter one survives into the next generation, as shown below.

$$x_{i,G+1} = \begin{cases} u_{i,G+1}, \ if\ f(u_{i,G+1}) \leq f(x_{i,G}) \\ x_{i,G}, \ if\ f(u_{i,G+1}) > f(x_{i,G}) \end{cases} \tag{4}$$

III. Weaknesses of UDE-III and Scope of Potential Improvements

In this section, the weaknesses of UDE-III are briefly discussed.

- **Strategy Pool** - In UDE-III, the strategy pool consists of DE/rand/1, DE/current-to-rand/1, and DE/current-to-$p$best/1 mutation strategies. The performance of UDE-III on several functions indicated that it may be converging prematurely. Thus, we realized that the exploration capability of the strategy pool may be a weakness of UDE-III and that the algorithm could benefit from enhancing the exploration power of the strategy pool.
- **Ranking-based mutation** - Inspired by the performance of the ranking-based parent selection for mutation proposed by Gong and Cai [12], it was incorporated in UDE-II and UDE-III. However, on further investigation, we observed that the ranking-based parent selection was leading to over-exploitation of fitter solutions and resulting in the inferior performance of the algorithm on several problems.
- **Parameter tuning** - Although most of the parameters in UDE-III were carefully tuned, the parameters related to the stagnation strategy were not optimally tuned.

IV. Proposed Unified Differential Evolution-IV

In this section, the proposed algorithm UDE-IV is presented. The proposed UDE-IV is a significantly enhanced version of UDE-III, which secured the 1st rank in the CEC 2024 competition on constrained real parameter optimization [7]. The components of UDE-IV are described in detail below.

A. Strategy Pool

The strategy pool in UDE-IV consists of DE/rand/1, DE/current-to-rand/1, DE/current-to-$p$best/1, and DE/rand-to-$p$best/1 trial vector generation strategies. The first three strategies are the same as in UDE III, but we have incorporated DE/rand-to-$p$best/1 in the strategy pool of UDE IV to improve the exploration capabilities of the algorithm.

B. Crossover

UDE-IV employs the binomial crossover operator for the 30D problems in the CEC 2025 competition just like UDE-III.

C. Dual Population

UDE-III uses three trial vector generation strategies - DE/rand/1, DE/current-to-rand/1, and DE/current-to-$p$best/1, whereas UDE-IV uses four trial vector generation strategies - DE/rand/1, DE/current-to-rand/1, DE/current-to-$p$best/1, and DE/rand-to-$p$best/1. At each generation, UDE-IV divides the current population into two sub-populations. In the top sub-population, it employs three trial vector generation strategies (DE/rand/1, DE/current-to-rand/1, and DE/current-to-$p$best/1) on each target vector. The bottom sub-population in UDE-IV applies only one trial vector generation strategy on each target vector. Specifically, the bottom sub-population employs strategy adaptation, wherein the probability of the three trial vector generation strategies implemented in the top sub-population is adapted in every generation by learning from their experiences in generating successful/unsuccessful trial vectors across the whole population in the previous learning period. To enhance the exploration capabilities, UDE-IV probabilistically uses DE/current-to-$p$best/1 and DE/rand-to-$p$best/1 in the bottom sub-population. Specifically, DE/current-to-$p$best/1 is implemented with a probability of $Prob$ while DE/rand-to-$p$best/1 is implemented with a probability of $1 - Prob$.

D. Strategy Adaptation

As discussed in the above sub-section, in the top sub-population A, DE/rand/1, DE/current-to-rand/1, and DE/current-to-$p$best/1 are deployed for each target vector to generate three trial vectors as in UDE III. The three trial vectors generated corresponding to each target vector are compared among themselves, and the trial vector generation strategy with the best trial vector scores a win. The other two trial vector generation strategies are assigned losses. In the bottom sub-population B, corresponding to each target vector, a trial vector generation strategy is probabilistically employed. If the trial vector is better than the corresponding target vector, it scores a win; otherwise, it scores a loss. The probability of employing a trial vector generation strategy is equal (i.e., 0.33) for the first Lp generations. Thereafter, in every generation, the probability of employing a trial vector generation strategy is calculated using the same method as proposed in SaDE. The probability evaluation method considers the number of wins and losses of each trial vector generation strategy over the whole population across the previous learning period Lp.

TABLE I. UDE IV RESULTS OBTAINED FOR $30D$ ($MaxFES = 20000 \times D$)

Func	F1	F2	F3	F4	F5	F6	F7
Best	0	0	0	0	0	0	-881.73
Mean	[000]	[000]	[000]	[000]	[000]	[000]	[000]
Worst	0	0	0	0	0	0	0
STD	5.24E-29	5.23E-29	5.40E-29	10.315	2.24E-28	60.962	-600.42
SR	2.51E-28	3.25E-28	2.02E-28	13.573	2.32E-27	349.07	-153.57
$V\bar{i}ol$	7.23E-29	6.90E-29	6.08E-29	5.9162	4.61E-28	116.47	181.63
c	100	100	100	100	100	100	100
$\bar{v}$	0	0	0	0	0	0	0
Func	F8	F9	F10	F11	F12	F13	F14
Best	-0.00028398	-0.0026655	-0.00010284	-4.2652	3.9825	0	1.4085
Mean	[000]	[000]	[000]	[100]	[000]	[000]	[000]
Worst	0	0	0	1.1006	0	0	0
STD	-0.00028398	-0.0026655	-0.00010284	-488.24	3.9892	2.21E-26	1.4085
SR	-0.00028398	-0.0026655	-0.00010284	-1730.3	4.148	2.13E-25	1.4085
$V\bar{i}ol$	9.93E-10	0	1.68E-10	676.32	0.03308	4.28E-26	4.88E-16
c	100	100	100	0	100	100	100
$\bar{v}$	0	0	0	62.069	0	0	0
Func	F15	F16	F17	F18	F19	F20	F21
Best	2.3561	0	0.032237	36.52	0	1.3144	3.9825
Mean	[000]	[000]	[100]	[000]	[100]	[000]	[000]
Worst	0	0	14.5	0	21375	0	0
STD	2.3561	0	0.32825	36.521	0	1.8213	7.1514
SR	2.3561	0	1.303	36.527	0	2.3233	14.603
$V\bar{i}ol$	5.63E-07	0	0.44044	0.001352	0	0.2944	3.9147
c	100	100	0	100	0	100	100
$\bar{v}$	0	0	14.5	0	21375	0	0
Func	F22	F23	F24	F25	F26	F27	F28
Best	6.92E-08	1.4085	2.3561	1.75E-10	0.12186	36.52	62.814
Mean	[000]	[000]	[000]	[000]	[100]	[000]	[100]
Worst	0	0	0	0	15.5	0	21455
STD	4.1082	1.4294	2.3561	2.8274	0.67051	36.751	95.636
SR	94.536	1.4954	2.3561	62.832	1.0456	39.218	123.66
$V\bar{i}ol$	18.857	0.037889	1.36E-15	12.599	0.34118	0.63564	21.037
c	100	100	100	100	0	100	0
$\bar{v}$	0	0	0	0	15.38	0	21454

E. Constraint Handling

The constraint handling technique in UDE-IV is same as that in UDE-III. It is inspired by the combination of ϵ constraint handling technique [17] and superiority of feasible (SOF) solutions method [18] proposed in C^{2}oDE [13]. The top sub-population in UDE-IV employs the same constraint handling technique as in C^{2}oDE [13] because the evolution of solutions in the top sub-population in UDE-IV is similar to that in C^{2}oDE. However, as the bottom sub-population generates only one trial vector corresponding to each target vector, only the ϵ-constraint handling technique [17] is employed in the bottom sub-population.

The parameter settings for the ϵ-constraint handling technique follow the same values as used in C^{2}oDE [13] except for the parameter cp. Our experimental investigations demonstrated that the value of cp may vary considerably as cp is derived from the following equation:

$$cp = -(log\epsilon_0 + \lambda)/(log(1-p) \tag{5}$$

Here, ϵ_0 is the initial threshold and is set to be the maximum degree of constraint violation of the initial population. The maximum degree of constraint violation of the initial population, i.e., ϵ_0, may vary significantly depending upon the problem characteristics. As a result, cp may also vary substantially. We observed that in some problems, cp may take very high values, because of which ϵ may decrease at a rapid exponential rate. This has a negative impact on the performance of UDE-IV on several problems. Our experimental investigation demonstrated that limiting cp to 33 significantly improved the performance of UDE-IV just like UDE-III. Hence, in UDE-IV, we implement the condition that if $cp \geq 33$ then cp = 33.

F. Parameter Adaptation

The parameter adaptation in UDE-IV is the same as in UDE-III. In particular, UDE-IV employs SHADE [11] style parameter adaptation. UDE-IV employs one pair of memories M_F and M_C for adaptation of F and CR corresponding to DE/rand/1/bin and DE/current-to-$p$best/1, and utilizes memory M_F for adaptation of F corresponding to DE/current-to-rand/1. However, the parameter adaptation proposed in SHADE [11] is suited only for bound-constrained, i.e., unconstrained optimization. Thus, UDE-IV employs the parameter adaptation as proposed in LSHADE44 [19], which extends the SHADE [11] style parameter adaptation to constrained optimization problems. It is noted that only the top sub-population is utilized for the parameter adaptation in UDE-IV, just like in UDE-III. This is a weakness of UDE-IV, which could be addressed in the future to further improve the algorithm.

G. Migration between Sub-populations

At the beginning of every generation, the population members are sorted according to the superiority of the feasible solutions method. This step leads to the migration of fitter (in terms of feasibility) solutions from bottom sub-population B to top sub-population A and the migration of less fit solutions from top sub-population A to bottom sub-population B. The advantage of migration is that the CoDE-style trial vector generation strategy (i.e., implementation of three strategies) always gets implemented on the top sub-population members. Since UDE-IV employs DE/current-to-rand/1 and DE/current-to-$p$best/1 trial vector generation strategies, the CoDE style trial vector generation strategy being implemented on the best population members (in terms of feasibility) can promote exploitation of the fitter feasible solutions, and lead to faster convergence. On the other hand, since less fit solutions are part of the bottom sub-population and UDE-IV employs DE/current-to-$p$best/1 in the bottom sub-population with a probability $Prob$ and DE/rand-to-$p$best/1 with a probability $1 - Prob$, it favors the exploration.

H. Stagnation Strategy

The stagnation strategy in UDE-IV is the same as in UDE-III. However, we still elaborate on it here for completeness.

To design an effective strategy to overcome stagnation, several important research questions need to be addressed. First, when can an individual be considered to be stagnated? Generally, if an individual has not been replaced by its trial vector for many generations, it can be said to be stagnated. Thus, we used a parameter SG, which indicates the number of generations for which, if an individual has not shown improvement, it can be said to be stagnated.

Second, should a stagnated individual be immediately replaced, or should the stagnation strategy be applied if a certain proportion of the population stagnates? We researched this extensively and observed that replacing a stagnated individual immediately does not result in consistent superior algorithm performance. Hence, we used a parameter $SProp$ to indicate the threshold for the proportion of the stagnated population beyond which the stagnation strategy is activated. For example, $SProp$ = 50% activates the stagnation strategy only when the number of stagnated individuals in the population exceeds 50% of population size NP.

Third, the choice of an individual to replace a stagnated individual is important. In UDE-IV, we maintain an external archive of size NP to store unsuccessful trial vectors generated during evolution. The external archive is updated at every generation, and the size of the archive is always maintained to be less than or equal to NP using the superiority of feasibility principle. Once the proportion of the stagnated population is greater than $SProp$ of NP, a stagnated individual is replaced in every generation with a randomly selected vector from the external archive. In this way, UDE-IV effectively utilizes the large number of unsuccessful trial vectors in the stagnation strategy.

I. Replacement

UDE-IV employs the traditional one-to-one comparison between the target vector and the trial vector to select a solution for the next generation. It is noted that, as discussed above in the constraint handling strategy, the one-to-one comparison and selection between the target vector and the trial vector is based on the ϵ constraint handling method.

J. Memory

Since UDE-IV employs a combination of ϵ-constraint handling technique [17] and superiority of feasible (SOF) solutions method [18], there is a possibility that the best solution in terms of SOF may get lost from the evolving population. Hence, UDE-IV utilizes an external memory, and at the end of every generation, the memory is updated with the best solution concerning SOF, similar to UDE III.

V. Modifications made in UDE-III to Design UDE-IV

In this section, we summarize the modifications made in UDE-III to design UDE-IV.

- **Strategy pool** - UDE-IV employs four trial vector generation strategies: DE/rand/1, DE/current-to-rand/1, DE/current-to-$p$best/1, and DE/rand-to-$p$best/1, unlike UDE-III, which employs only the first three strategies. The top sub-population applies DE/rand/1, DE/current-to-rand/1, and DE/current-to-$p$best/1 to each target vector. The bottom sub-population employs a strategy adaptation mechanism and utilizes a single strategy per target vector. The strategy adaptation mechanism dynamically adjusts the probabilities of the three trial vector generation strategies based on their past success in generating promising solutions. To further boost exploration capabilities compared to UDE-III, UDE-IV probabilistically uses DE/current-to-$p$best/1 and DE/rand-to-$p$best/1 within the bottom sub-population as:

$$\begin{cases} \text{DE/current-to-}p\text{best/1} & \text{if } rand < Prob, \\ \text{DE/rand-to-}p\text{best/1} & \text{otherwise.} \end{cases} \tag{6}$$

TABLE II. COMPARISON OF UDE-IV WITH UDE-III ON $30D$ PROBLEMS.

Func	UDE III	UDE IV
	Mean $\pm$ STD	Mean $\pm$ STD
F1	1.570E-28 $\pm$ 9.900E-29	5.237E-29 $\pm$ 7.232E-29
F2	1.390E-28 $\pm$ 9.490E-29	5.233E-29 $\pm$ 6.901E-29
F3	9.221E+01 $\pm$ 5.650E+01	5.395E-29 $\pm$ 6.084E-29
F4	6.515E+00 $\pm$ 6.921E+00	1.032E+01 $\pm$ 5.916E+00
F5	1.595E-01 $\pm$ 7.973E-01	2.241E-28 $\pm$ 4.614E-28
F6	0.000E+00 $\pm$ 0.000E+00	6.096E+01 $\pm$ 1.165E+02
F7	-6.732E+02 $\pm$ 1.517E+02	-6.004E+02 $\pm$ 1.816E+02
F8	-2.840E-04 $\pm$ 4.920E-12	-2.840E-04 $\pm$ 9.930E-10
F9	-2.666E-03 $\pm$ 0.000E+00	-2.666E-03 $\pm$ 0.000E+00
F10	-1.028E-04 $\pm$ 1.800E-14	-1.028E-04 $\pm$ 1.680E-10
F11	(28%) $\pm$	(0%) $\pm$ 6.207E+01
F12	3.992E+00 $\pm$ 2.225E-02	3.989E+00 $\pm$ 3.308E-02
F13	4.784E-01 $\pm$ 1.322E+00	2.207E-26 $\pm$ 4.284E-26
F14	1.409E+00 $\pm$ 4.350E-16	1.409E+00 $\pm$ 4.882E-16
F15	2.356E+00 $\pm$ 1.440E-06	2.356E+00 $\pm$ 5.627E-07
F16	0.000E+00 $\pm$ 0.000E+00	0.000E+00 $\pm$ 0.000E+00
F17	(0%) $\pm$ 1.471E+01	(0%) $\pm$ 1.450E+01
F18	3.655E+01 $\pm$ 1.394E-01	3.652E+01 $\pm$ 1.352E-03
F19	(0%) $\pm$ 2.138E+04	(0%) $\pm$ 2.138E+04
F20	1.851E+00 $\pm$ 2.883E-01	1.821E+00 $\pm$ 2.944E-01
F21	9.280E+00 $\pm$ 8.507E+00	7.151E+00 $\pm$ 3.915E+00
F22	2.556E+01 $\pm$ 4.944E+01	4.108E+00 $\pm$ 1.886E+01
F23	1.450E+00 $\pm$ 4.432E-02	1.429E+00 $\pm$ 3.789E-02
F24	2.356E+00 $\pm$ 1.080E-07	2.356E+00 $\pm$ 1.360E-15
F25	2.513E-01 $\pm$ 1.257E+00	2.827E+00 $\pm$ 1.260E+01
F26	(0%) $\pm$ 1.528E+01	(0%) $\pm$ 1.538E+01
F27	3.770E+01 $\pm$ 3.241E+00	3.675E+01 $\pm$ 6.356E-01
F28	(0%) $\pm$ 2.144E+04	(0%) $\pm$ 2.145E+04
$+/\approx/-$	–	13/8/7

- **Ranking-based selection** - The ranking-based selection for mutation has been removed in UDE-IV. With the exclusion of ranking-based selection, the performance of UDE-IV improved significantly on several problems. The reason may be that the usage of ranking-based selection was leading to over-exploitation.
- **Parameter tuning** - In UDE IV, we tuned the parameters related to the stagnation strategy, namely SG and $SProp$, since these were not optimally tuned in UDE-III.

VI. EXPERIMENTAL RESULTS

The proposed algorithm UDE-IV is tested on the 28 CEC 2025 benchmark-constrained optimization problems with 30 dimensions (D). The parameter settings of UDE-IV are as follows:

- Population size NP = 100,
- Size of top sub-population T = 25,
- Learning period Lp = 25 generations,
- SG = 25,
- $SProp$ = 40%
- $Prob$ = 0.75

On each problem, 25 independent runs of UDE-IV are taken. The results obtained on $30D$ problems are reported in Table I. The comparative results of UDE-IV and UDE-III are reported in Table II. Table II shows that UDE-IV is inferior, similar, and superior to UDE-III on 7, 8, and 13 functions, respectively. Overall, this shows the effectiveness of UDE-IV in comparison to UDE-III in solving constrained real-parameter optimization problems.

VII. CONCLUSION

In this paper, an enhanced unified differential evolution algorithm, named UDE-IV, has been proposed to solve real parameter-constrained optimization problems of the CEC 2025 competition. UDE-IV is a significantly enhanced version of UDE-III, which secured the 1st rank in the CEC 2024 competition on constrained real parameter optimization. UDE-IV is built on the philosophy of unified algorithm design, comprehensive offline + online learning, and black-box thinking. UDE-IV has been designed by targeting the weaknesses of UDE III (i.e., by implementing black box thinking) and by focusing on each and every design element (i.e., by focusing on unified algorithm design). It leverages many state-of-the-art algorithms proposed in the literature (i.e., offline learning). Moreover, it implements online learning in the form of strategy adaptation, parameter adaptation, and the use of unsuccessful trial vectors to overcome stagnation. Specifically, compared to UDE-III, UDE-IV improves by excluding the ranking-based mutation, enhancing the exploration capabilities of the strategy pool, and effective parameter tuning of the stagnation strategy. The comparative experimental results on the CEC 2025 test suite against UDE-III demonstrate the superiority of UDE-IV in solving real parameter-constrained optimization problems.

It is worthwhile noting that only the top sub-population is utilized for the parameter adaptation in UDE-IV just like in UDE-III. This is a weakness of UDE-IV which could be addressed in the future to further improve the algorithm.

REFERENCES

[1] R. Storn and K. Price, "Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces," International Computer Science Institute, Berkeley, CA, USA, 1995, TR-95–012, Tech. Rep., 1995.

[2] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4 – 31, 2011.

[3] S. Das, S. S. Mullick, and P. Suganthan, "Recent advances in differential evolution – an updated survey," *Swarm and Evolutionary Computation*, pp. –, 2016.

[4] D. Chauhan, A. Trivedi, and Shivani, "A multi-operator ensemble lshade with restart and local search mechanisms for single-objective optimization," *arXiv preprint arXiv:2409.15994*, 2024.

[5] D. Chauhan, Shivani, D. Jung, and A. Yadav, "Advancements in multimodal differential evolution: A comprehensive review and future perspectives," *arXiv preprint arXiv:2504.00717*, 2025.

[6] A. Trivedi, D. Srinivasan, and N. Biswas, "An improved unified differential evolution algorithm for constrained optimization problems," IEEE Congress on Evolutionary Computation Conference 2018, Tech. Rep., 2018.

[7] A. Trivedi and D. Chauhan, "UDE-III: An enhanced unified differential evolution algorithm for constrained optimization problems," *arXiv preprint arXiv:2410.03992*, 2024.

[8] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, Oct 2009.

[9] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, Feb 2011.

[10] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, April 2009.

[11] R. Tanabe and A. Fukunaga, "Evaluating the performance of SHADE on CEC 2013 benchmark problems," in *2013 IEEE Congress on Evolutionary Computation*, June 2013, pp. 1952–1959.

[12] W. Gong and Z. Cai, "Differential evolution with ranking-based mutation operators," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2066–2081, Dec 2013.

[13] B. C. Wang, H. X. Li, J. P. Li, and Y. Wang, "Composite differential evolution for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–14, 2018.

[14] Z. Zeng, Z. Hong, H. Zhang, M. Zhang, and C. Chen, "Improving differential evolution using a best discarded vector selection strategy," *Inf. Sci.*, vol. 609, pp. 353–375, Sep. 2022.

[15] M. Syed, *Black box thinking: Marginal Gains and the Secrets of High Performance*. John Murray Publishers Ltd., London, 2016.

[16] Q. Kangjia, W. Xupeng, B. Xuanxuan, C. Peng, P. Kenneth V., S. Ponnuthurai N., L. Jing, W. Guohua, and Y. Caitong, "Evaluation criteria for CEC 2024 competition and special session on numerical optimization considering accuracy and speed," Zhengzhou University, Central South University, Henan Institute of Technology, Qatar University, Tech. Rep., 2018.

[17] T. Takahama and S. Sakai, "Constrained optimization by the ϵ constrained differential evolution with an archive and gradient-based mutation," in *IEEE Congress on Evolutionary Computation*, July 2010, pp. 1–9.

[18] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 311 – 38, 2000.

[19] R. Poláková, "L-SHADE with competing strategies applied to constrained optimization," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 1683–1689.