E^2-CTS-DE: Chaotic Thompson Sampling for Operator Selection in Differential Evolution

Luis A. Beltrán,
Omar Alvarez, Diego Oliva,
Javier Galvis-Chacón, Mario A. Navarro
Universidad de Guadalajara, CUCEI
Guadalajara, México
{luis.beltran7882, omar.alvarez7934,
javier.galvis8583}@alumnos.udg.mx,
diego.oliva@cucei.udg.mx
mario.navarro@academicos.udg.mx

Daniel F. Zambrano-Gutierrez
Tecnologico de Monterrey,
Monterrey, Mexico
A00836756@exatec.tec.mx

Saúl Zapotecas-Martínez
Instituto Nacional de Astrofisica
Optica y Electronica
Tonantzintla, Puebla, México
szapotecas@inaoep.mx

***Abstract*—Differential Evolution (DE) is highly sensitive to the choice of search operators and to the exploration–exploitation balance induced by their interaction. Many DE variants rely on fixed operator configurations or handcrafted adaptation rules, which can reduce robustness on heterogeneous and deceptive landscapes. This work formulates operator adaptation from a selection hyper-heuristic viewpoint and proposes E^2-CTS-DE (Exploration–Exploitation Chaotic Thompson Sampling Differential Evolution), a learning-driven DE variant that adaptively selects mutation–crossover combinations during the search. The method models each mutation–crossover pair as a low-level heuristic and employs a Multi-Armed Bandit scheme driven by Thompson Sampling to perform online selection based on improvement feedback. In addition, the initial population is generated via chaotic maps to enhance diversity and strengthen early exploration. Experiments on a standard suite of continuous black-box benchmark functions show that E^2-CTS-DE is competitive and frequently outperforms state-of-the-art DE variants and representative population-based optimizers, achieving robust performance across multimodal, unimodal, composite, and shifted landscapes.**

***Index Terms*—Differential Evolution, Selection Hyper-Heuristic, Multi-Armed Bandit, Thompson Sampling, Chaotic Initialization, Exploration–Exploitation Trade-off**

I. Introduction

Differential Evolution (DE) is a widely used evolutionary algorithm for continuous optimization due to its simplicity, robustness, and competitive performance across a broad range of problems [1]. However, its effectiveness depends strongly on the choice of mutation and crossover operators, as well as on maintaining an appropriate balance between exploration and exploitation during the search process. Fixed operator configurations or static adaptation schemes may lead to premature convergence or inefficient search behavior when dealing with complex fitness landscapes.

From a higher-level perspective, the problem of selecting suitable search operators naturally aligns with the selection hyper-heuristic paradigm, which aims to dynamically choose among a set of low-level heuristics based on their observed performance using problem-independent feedback [2], [3]. While selection hyper-heuristics have been extensively studied, most existing approaches target combinatorial optimization problems [2], and rely on problem-specific neighborhood moves, making their direct application to population-based continuous optimization algorithms nontrivial.

Recent advances have shown that learning-based decision models can be effectively embedded within evolutionary algorithms, leading to domain-specific selection hyper-heuristics that operate over evolutionary operators [4], [5]. In this context, the Multi-Armed Bandit (MAB) framework provides a principled mechanism for adaptive operator selection by explicitly addressing the exploration–exploitation trade-off [6]. Among MAB strategies, Thompson Sampling (TS) has attracted significant attention due to its Bayesian formulation [7], computational efficiency, and strong empirical performance.

In parallel, chaotic population initialization strategies have been proposed to enhance early-stage exploration in evolutionary algorithms. The nonlinear properties of chaotic maps enable improved coverage of the search space compared to purely random initialization [8]. Nevertheless, chaotic initialization alone does not provide adaptive control over the search dynamics throughout the evolutionary process.

Motivated by these observations, this work proposes E^2-CTS-DE (Exploration–Exploitation Chaotic Thompson Sampling Differential Evolution), a learning-based selection hyper-heuristic embedded within Differential Evolution. In E^2-CTS-DE, combinations of mutation and crossover operators are treated as low-level heuristics, while a high-level hyper-heuristic layer based on a Multi-Armed Bandit model guided by Thompson Sampling adaptively selects the most suitable operator combinations using a problem-independent success signal. Chaotic maps are employed to generate a diverse initial population, strengthening early exploration, while the hyper-heuristic mechanism continuously adapts operator selection to balance exploration and exploitation throughout the search.

Extensive experimental results on standard benchmark func-

TABLE I
MUTATION VARIANTS CONSIDERED IN THIS WORK (GROUPED BY SEARCH BEHAVIOR) [1], [10]–[13].

ID	Variant	Type	Core idea
M1	`DE/rand/1`	E	Random base + one difference (baseline).
M2	`DE/rand/2`	E	Two differences to increase diversity.
M3	`DE/current-to-rand/1`	E	Current drifts toward random + differential term.
M4	Gaussian-based	X	Gaussian adaptation/perturbation to reduce stagnation.
M5	`DE/best/1`	X	Best as base to accelerate convergence.
M6	Trigonometric	X	Trigonometric weighting to balance diversity/convergence.
M7	Reflection-based	X	Opposition/reflection learning for faster progress.
M8	`DE/current-to-pbest/1`	X	Guided by a top p-best individual.
M9	Hemostasis-based	X	Bio-inspired scaling modulation for stable convergence.
M10	`DE/best/2`	X	Best-based with two differences (intensification).

Type: E = exploration-focused, X = exploitation/convergence-focused.

TABLE II
CROSSOVER MECHANISMS USED IN THE MULTI-CROSSOVER SCHEME.

ID	Method	Mechanism	Ref.
C1	Binomial	Per-dimension exchange with crossover probability.	[9]
C2	Exponential	Copies a contiguous block from mutant to trial.	[9]
C3	Continuous	Real-coded interpolation between parents.	[14]
C4	Directional	Biases recombination toward improving directions.	[15]
C5	SBX	Simulated binary crossover for real variables.	[16]

tions demonstrate that E^2-CTS-DE achieves competitive and, in many cases, superior performance compared to state-of-the-art Differential Evolution variants, highlighting the effectiveness of integrating chaotic dynamics with Bayesian learning-based selection hyper-heuristics for continuous optimization.

II. BACKGROUND

A. Differential Evolution

Differential Evolution (DE) is a population-based evolutionary algorithm that generates candidate solutions by perturbing existing individuals through scaled vector differences and recombining them to form trial vectors [9]. Given an initial population, each generation applies three operators: *mutation* constructs a donor vector, *crossover* blends the donor and target vectors to produce a trial solution, and *selection* retains the better of the trial and target based on the objective function.

The search behavior of DE is strongly determined by the mutation scheme, i.e., the choice of base vector and the number/type of differential terms. In this work, ten mutation variants are considered and grouped into exploration-focused and exploitation/convergence-focused strategies (Table I). In short, exploration-oriented rules rely on entirely random bases and/or additional random differences to increase diversity. In contrast, exploitation-oriented variants incorporate guidance from the current best or elite individuals (e.g., `best` or p-`best`) and/or adaptive mechanisms (e.g., Gaussian, reflection/opposition, or bio-inspired modulation) to intensify the search and stabilize convergence.

Moreover, crossover controls how information from the donor vector is injected into the target solution. Instead of fixing a single recombination rule, we employ a multi-crossover pool comprising five mechanisms (see Table II). At each application, the crossover operator is selected based on its historical performance, allowing the system to adaptively favor the most effective recombination pattern for the current search regime (e.g., dimension-wise mixing, block-wise inheritance, interpolation-based recombination, or directionally biased mixing).

B. Multi-Armed Bandits and Thompson Sampling

The Multi-Armed Bandit (MAB) problem was introduced in [17] to model sequential decision-making under uncertainty. Consider a finite set of arms $\mathcal{A} = \{1, \dots, K\}$. At each time step $t = 1, 2, \dots$, an agent selects an arm $A_t \in \mathcal{A}$ and observes a stochastic reward $R_t \in \mathbb{R}$ drawn from an unknown distribution ν_{A_t}. In the classical stationary formulation, rewards are assumed i.i.d. over time for each arm and independent across arms [18]. The objective is to maximize the expected cumulative reward ($\max\ \mathbb{E}\left[\sum_{t=1}^{T} R_t\right]$), which induces the exploration-exploitation trade-off. The agent must allocate trials to infer which arm is optimal while preferentially selecting arms with higher observed performance [18].

Thompson Sampling (TS) is a Bayesian probability-matching policy for MABs [18]. TS maintains, for each arm, a posterior distribution over its expected reward and selects actions according to the posterior probability of being optimal. At each time step, TS samples one parameter from each posterior and selects the arm with the maximum sample. In the Bernoulli bandit setting, each arm i yields $R_t \in \{0, 1\}$ with unknown success probability p_i. Using the conjugate Beta prior $p_i \sim \mathrm{Beta}(\alpha_i, \beta_i)$ [7], the density is

$$\mathrm{Beta}(\alpha, \beta): \quad p(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \quad (1)$$

where $\alpha > 0$, $\beta > 0$, and $\Gamma(\cdot)$ denotes the Gamma function. A common initialization is the non-informative prior $\mathrm{Beta}(1, 1)$ on $[0, 1]$.

Let S_i and F_i denote the cumulative numbers of successes and failures observed for arm i, respectively. After each observation, the posterior updates as

$$p_i \mid \mathcal{D} \sim \mathrm{Beta}(\alpha_i + S_i,\ \beta_i + F_i). \quad (2)$$

Then, at time t, TS draws

$$\theta_i \sim \mathrm{Beta}(\alpha_i + S_i,\ \beta_i + F_i), \quad i = 1, \dots, K, \quad (3)$$

and selects

$$A_t = \arg \max_{i \in \{1, \dots, K\}} \theta_i. \quad (4)$$

This randomized selection mechanism yields a principled balance between exploration and exploitation through posterior uncertainty, while remaining simple to implement and effective in practice [18].

C. Automated Algorithm Design and Hyper-heuristic

Designing Metaheuristic (MH) optimizers typically involves selecting algorithmic components and setting their associated hyperparameters [19]. Since this design space can be combinatorially ample, manual trial-and-error often explores only a small fraction of feasible configurations, potentially leading to suboptimal designs. Automated Algorithm Design and Configuration (AADC) addresses this limitation by casting algorithm construction and tuning as an optimization problem over a predefined design space [20].

In this work, the algorithmic template is DE, whose search dynamics are largely governed by the choice of mutation and crossover operators. Let $\mathcal{M}$ be the set of available mutation schemes (Table I) and $\mathcal{C}$ the set of crossover mechanisms (Table II). The AAD design space is defined as the Cartesian product

$$\mathcal{A} = \mathcal{M} \times \mathcal{C}, \qquad |\mathcal{A}| = |\mathcal{M}| \cdot |\mathcal{C}|, \tag{5}$$

where each design (action) $a = (M_k, C_m) \in \mathcal{A}$ specifies a mutation–crossover pair applied inside the DE cycle. Thus, instead of fixing a single operator pair for the entire run, the design goal is to construct an adaptive DE instance by defining a selection policy over $\mathcal{A}$.

One way to address this AAD setting is through a Hyper-Heuristic (HH) scheme [21] in which a high-level controller, at each step, selects an action $a \in \mathcal{A}$ from a finite operator library. In this work, the controller is modeled as a MAB and implemented with Thompson Sampling; each arm (M_k, C_m) is updated using Bernoulli improvement feedback, yielding an explicit exploration-exploitation trade-off. Once the operator library is fixed, model-based tuning can further refine operator and/or controller hyperparameters [22].

D. Chaotic Initialization

Random initialization may yield clustered starting points, which can slow early progress and increase the risk of premature convergence. To strengthen initial diversity, the population is generated using a pool of ten chaotic maps. For each individual i and dimension d, one map is selected at random to produce a normalized chaotic value $\psi_{i,d} \in (0,1)$, which is then scaled to the variable bounds:

$$X^c_{i,d} = X^{\min}_d + \psi_{i,d}\left(X^{\max}_d - X^{\min}_d\right), \tag{6}$$

where $X^{\min}_d$ and $X^{\max}_d$ denote the lower and upper bounds of decision variable d, respectively. The chaotic maps used to generate $\psi_{i,d}$ are summarized in Table III. This procedure yields a more diverse initial population and improves the coverage of the search space at the start of each run.

III. Methodology

E[2]-CTS-DE is a DE variant that couples chaotic population initialization with an online operator-selection mechanism modeled as a MAB and implemented via TS. The resulting feedback loop is summarized in Fig. 1. Each run starts by generating the initial population using ten chaotic maps (Chebyshev, Circle, Gauss/Mouse, Iterative, Logistic, Piecewise, Sine, Singer, Sinusoidal, and Tent) to promote diversity. For decision-variable bounds $[L_d, U_d]$, $d = 1, \ldots, D$, normalized chaotic values $z_{i,d} \in (0,1)$ are scaled as

$$x^{(0)}_{i,d} = L_d + z_{i,d}\,(U_d - L_d), \qquad i = 1, \ldots, NP,\ d = 1, \ldots, D, \tag{7}$$

using a fixed random seed to generate the chaotic initial conditions and any auxiliary random choices.

After initialization, DE evolves a population $\mathbf{X}^{(g)} = \{\mathbf{x}^{(g)}_1, \ldots, \mathbf{x}^{(g)}_{NP}\}$ with $NP = 50$ on a black-box minimization problem $(\mathfrak{X}, f)$, where $\mathfrak{X} \subset \mathbb{R}^D$ and $f : \mathfrak{X} \to \mathbb{R}$. The operator library comprises 10 mutation variants and 5 crossover mechanisms (Section II-A), defining an action space $\mathcal{A} = \mathcal{M} \times \mathcal{C}$ with $|\mathcal{A}| = 50$. Each arm is a mutation–crossover pair

$$a_s = (M_s, C_s), \qquad M_s \in \{1, \ldots, 10\}, \quad C_s \in \{1, \ldots, 5\}. \tag{8}$$

All arms are initialized with the same non-informative Beta prior, $\mathrm{Beta}(1,1)$.

TS operates at the individual level. For each target vector $\mathbf{x}^{(g)}_i$, a single arm a_s is selected and applied exclusively to generate the donor and trial vectors for that individual, so different individuals within the same generation may use different operator pairs. For each arm $a \in \mathcal{A}$, TS maintains success and failure counts (S_a, F_a) and samples

$$\theta_a \sim \mathrm{Beta}(S_a + 1,\ F_a + 1), \qquad a \in \mathcal{A}, \tag{9}$$

selecting the arm with the maximum sampled value. Given $a_s = (M_s, C_s)$, mutation M_s produces a donor vector $\mathbf{v}^{(g)}_i$ and crossover C_s recombines $\mathbf{x}^{(g)}_i$ and $\mathbf{v}^{(g)}_i$ to obtain the trial vector $\mathbf{u}^{(g)}_i$. DE then applies one-to-one selection for minimization:

$$\mathbf{x}^{(g+1)}_i = \begin{cases} \mathbf{u}^{(g)}_i, & \text{if } f(\mathbf{u}^{(g)}_i) < f(\mathbf{x}^{(g)}_i), \\ \mathbf{x}^{(g)}_i, & \text{otherwise.} \end{cases} \tag{10}$$

TABLE III
Chaotic maps used for population initialization [23].

ID	Map	Update rule
1	Chebyshev	$\psi_{t+1} = \cos\left(\kappa \cos^{-1}(\psi_t)\right)$
2	Circle	$\psi_{t+1} = \mathrm{mod}\left(\psi_t + 0.2 - \frac{0.5}{2\pi}\sin(2\pi\psi_t),\ 1\right)$
3	Iterative	$\psi_{t+1} = \sin\left(\frac{0.7\pi}{\psi_t}\right)$
4	Sine	$\psi_{t+1} = \sin(\pi\psi_t)$
5	Singer	$\psi_{t+1} = \mu\left(7.86\psi_t - 23.31\psi_t^2 + 28.75\psi_t^3 - 13.302875\psi_t^4\right),\ \mu = 1.07$
6	Sinusoidal	$\psi_{t+1} = 2.3\,\psi_t^2 \sin(\pi\psi_t)$
7	Tent	$\psi_{t+1} = \begin{cases} \psi_t/0.7, & \psi_t < 0.7, \\ \frac{10}{3}(1-\psi_t), & \psi_t \geq 0.7, \end{cases}$
8	Piecewise	$\psi_{t+1} = \begin{cases} \psi_t/P, & 0 \leq \psi_t < P, \\ (\psi_t - P)/(0.5 - P), & P \leq \psi_t < 0.5, \\ (1 - P - \psi_t)/(0.5 - P), & 0.5 \leq \psi_t < 1 - P, \\ (1-\psi_t)/P, & 1 - P \leq \psi_t < 1, \end{cases}$
9	Logistic	$\psi_{t+1} = 4\psi_t(1-\psi_t)$
10	Gauss/Mouse	$\psi_{t+1} = \begin{cases} 0, & \psi_t = 0, \\ \frac{1}{\psi_t} - \left\lfloor \frac{1}{\psi_t} \right\rfloor, & \text{otherwise,} \end{cases}$

Notes: $\psi_t \in (0,1)$ is the chaotic state at iteration t. Parameters κ (Chebyshev) and P (Piecewise) follow the standard settings used in the corresponding map definitions.

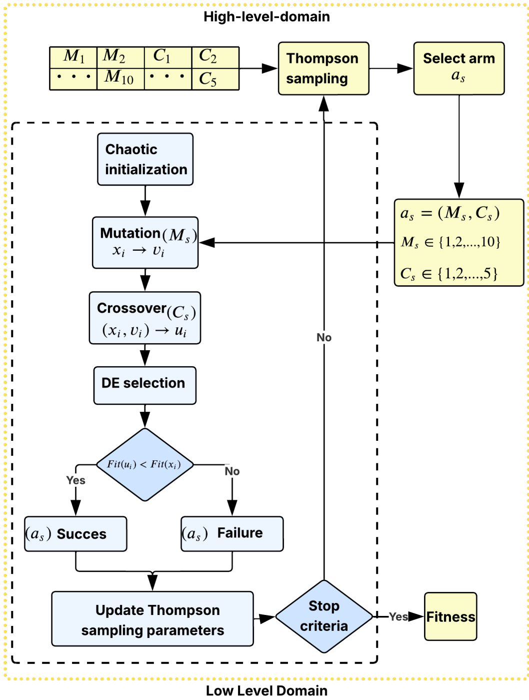


Fig. 1. Workflow of the proposed E^2-CTS-DE. Thompson Sampling selects, at the individual level, a mutation–crossover arm, which is applied within the DE cycle and updated using Bernoulli improvement feedback until the stopping criterion is met.

The same comparison defines the Bernoulli reward used to update the bandit model, consistent with the success/failure branch in Fig. 1:

$$r_i = \begin{cases} 1, & \text{if } f(\mathbf{u}_i^{(g)}) < f(\mathbf{x}_i^{(g)}), \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Only the selected arm is updated immediately after observing r_i:

$$S_{a_s} \leftarrow S_{a_s} + r_i, \qquad F_{a_s} \leftarrow F_{a_s} + (1 - r_i). \tag{12}$$

This online update biases sampling toward operator pairs that yield improvements while preserving exploration through posterior uncertainty.

A. Experimental Setup

All experiments were conducted in MATLAB R2022b on a benchmark suite of 32 continuous optimization functions [24] under a black-box minimization setting. The suite comprises 14 multimodal, 5 unimodal, 4 composite, and 9 shifted functions. For each function and dimension $D \in \{2, 10, 20, 30\}$, 30 independent runs were executed using a fixed population size $NP = 50$ and a budget of 100,000 objective-function evaluations (2,000 generations). For E^2-CTS-DE, DE control parameters were kept global and fixed across all arms to isolate the effect of the bandit-driven operator selection, using $F = 0.8$ and $CR = 0.2$. Performance is reported as the mean best-so-far value (AB), and its standard deviation (SD) across runs, and statistical differences among methods are assessed using the Friedman non parametric test [25]. We compare E^2-CTS-DE with DE [9], JADE [26], BDE [27], LSHADE [28], PSO [29], TACPSO [30], CMA-ES [31], CSA [32], KOA [33], EHO [34], DMO [35] and DSA [36]; baseline parameters followed the original references.

IV. Experimental results

A. Benchmark Comparison

Table IV summarizes the performance of E^2-CTS-DE and the reference algorithms on 32 benchmark functions (14 multimodal, 5 unimodal, 4 composite, and 9 shifted). For each function, Average Best (AB) and standard deviation (SD) are reported over 30 runs; the best and second-best AB values are highlighted in black and blue, respectively. On the multimodal set, E^2-CTS-DE ranks first in 10/14 functions and second in F2 and F9. Although it is not the top method on F12 and F13, it remains competitive with comparable AB values and no marked loss of stability, consistent with the enhanced exploration induced by chaotic initialization and Thompson Sampling-based operator selection. In the unimodal functions, the algorithm achieves the best performance across all evaluated cases, demonstrating intense exploitation; in these scenarios, the operator selection mechanism progressively converges toward more effective combinations, reinforcing search intensification and stable convergence. For the composite functions, E^2-CTS-DE achieves first place in two out of the four evaluated functions; in F21 and F22, it does not reach the top positions, although the obtained AB values are comparable to those of the leading methods, suggesting an adequate balance between exploration and exploitation in heterogeneous search landscapes. Finally, in the shifted function set, the proposed algorithm achieves first place in all nine evaluated functions, highlighting its robustness against changes in the global optimum location, which results from the combination of initial diversity and the dynamic operator adaptation induced by Thompson Sampling. In comparison, algorithms such as JADE and CMA-ES exhibit competitive, stable behavior across several scenarios, whereas other methods show performance that is more dependent on the function type. Overall, these results indicate that E^2-CTS-DE achieves an effective exploration–exploitation balance, maintaining robust and consistent performance across different search landscapes.

The convergence curves shown in Fig. 2 illustrate the behavior of the evaluated algorithms on representative benchmark functions with different search landscape characteristics, including multimodal (Ackley), unimodal (Rotated hyper-ellipsoid), composite (Composite 1), and shifted (Shifted Ackley) functions. Overall, E^2-CTS-DE exhibits faster, more stable convergence than the reference methods, achieving lower fitness values with fewer objective function evaluations. During the early stages of the search, the algorithm shows a pronounced reduction in fitness, reflecting compelling exploration of the search space. In contrast, in later stages, the convergence becomes more stable, evidencing a consistent exploitation process.

TABLE IV

AVERAGE BEST VALUE (AB) AND STANDARD DEVIATION (SD) OVER 30 RUNS FOR THE 32 BENCHMARK FUNCTIONS.

Function		E^2-CTS-DE	DE	JADE	BDE	LSHADE	PSO	TACPSO	CMAES	DSA	DMO	KOA	EHO
F1.Ackley	AB	**1.30×10^{-15}**	3.85×10^{-5}	2.45	2.18×10^{-4}	**9.81×10^{-14}**	1.38×10^{-7}	9.37×10^{-1}	1.29	8.85×10^{-7}	2.18	3.03×10^{-12}	2.88
	SD	**1.74×10^{-15}**	1.71×10^{-4}	4.54×10^{-1}	7.12×10^{-5}	**5.74×10^{-14}**	1.47×10^{-7}	8.29×10^{-1}	4.91	5.38×10^{-7}	3.09×10^{-1}	3.24×10^{-12}	9.86×10^{-1}
F2.Dixon	AB	**2.68×10^{-1}**	6.79×10^{-1}	6.67×10^{-1}	**3.29×10^{-1}**	6.67×10^{-1}	7.21×10^{-1}	1.96×10^{1}	6.67×10^{-1}	8.63×10^{-1}	5.38×10^{2}	6.67×10^{-1}	8.06×10^{-1}
	SD	1.25×10^{-1}	1.51×10^{-2}	1.09×10^{-4}	2.84×10^{-1}	**8.99×10^{-17}**	2.07×10^{-1}	6.23×10^{1}	9.68×10^{-6}	4.35×10^{-1}	1.86×10^{2}	**1.30×10^{-13}**	3.74×10^{-1}
F3.Griewank	AB	**0**	1.75×10^{-8}	4.34×10^{-4}	2.25×10^{-3}	**0**	9.02×10^{-3}	2.00×10^{-2}	2.27×10^{-15}	2.79×10^{-10}	1.03	4.27×10^{-3}	1.95×10^{-2}
	SD	**0**	3.13×10^{-8}	2.24×10^{-3}	5.39×10^{-3}	**0**	1.10×10^{-2}	2.02×10^{-2}	1.74×10^{-15}	9.47×10^{-10}	1.62×10^{-2}	7.92×10^{-3}	2.59×10^{-2}
F4.Infinity	AB	**0**	4.46×10^{-28}	2.76×10^{-28}	1.80×10^{-26}	**7.52×10^{-69}**	5.13×10^{-28}	3.31×10^{-55}	3.77×10^{-55}	1.16×10^{-40}	1.72×10^{-3}	6.12×10^{-58}	6.41×10^{-28}
	SD	**0**	3.18×10^{-28}	8.94×10^{-28}	3.50×10^{-26}	**1.42×10^{-68}**	1.31×10^{-27}	1.09×10^{-54}	9.54×10^{-55}	3.05×10^{-40}	1.05×10^{-3}	2.25×10^{-57}	1.44×10^{-27}
F5.Levy	AB	**1.50×10^{-32}**	2.65×10^{-11}	1.51×10^{-2}	1.44×10^{-8}	**1.10×10^{-26}**	1.73×10^{-1}	4.05×10^{-1}	4.62×10^{1}	2.37×10^{-13}	8.13	2.89×10^{-1}	2.32
	SD	**1.11×10^{-47}**	1.21×10^{-11}	8.29×10^{-2}	1.48×10^{-8}	**1.32×10^{-26}**	5.27×10^{-1}	3.52×10^{-1}	1.54×10^{1}	2.97×10^{-13}	2.20	3.12×10^{-1}	1.49
F6.Mishra 1	AB	**2.00**	**2.00**	**2.00**	2.01	**2.00**	5.28×10^{1}	**2.00**	**2.00**	**2.00**	8.87×10^{3}	**2.00**	1.02×10^{1}
	SD	**0**	**0**	**0**	2.10×10^{-3}	**0**	1.57×10^{2}	**0**	**0**	**0**	2.92×10^{4}	**0**	1.14
F7.Powell	AB	**2.44×10^{-53}**	1.24×10^{1}	2.17×10^{-4}	1.74×10^{-3}	**8.91×10^{-10}**	1.59×10^{-3}	5.01	1.22×10^{-2}	2.06×10^{-2}	4.62×10^{2}	1.06×10^{-4}	1.50×10^{-3}
	SD	**1.34×10^{-52}**	4.20	8.13×10^{-4}	1.09×10^{-3}	**1.69×10^{-9}**	7.18×10^{-4}	2.75×10^{1}	8.31×10^{-3}	1.02×10^{-2}	1.81×10^{2}	6.74×10^{-5}	5.52×10^{-4}
F8.Rastringin	AB	**0**	5.05×10^{1}	4.92	1.79	9.25×10^{-3}	4.11×10^{1}	6.13×10^{1}	1.60×10^{2}	**5.85×10^{-3}**	2.20×10^{2}	3.44×10^{1}	5.76×10^{1}
	SD	**0**	5.33	1.44	1.53	**9.48×10^{-3}**	1.31×10^{1}	1.28×10^{1}	6.99	1.17×10^{-2}	1.33×10^{1}	5.40	3.02×10^{1}
F9.Rosenbrock	AB	**9.49×10^{-1}**	2.99×10^{1}	2.30×10^{1}	**4.73**	9.45	3.42×10^{1}	2.73×10^{1}	9.51	3.62×10^{1}	1.60×10^{3}	1.53×10^{1}	3.95×10^{1}
	SD	5.20	4.85	1.33×10^{1}	9.40	**1.17**	2.56×10^{1}	3.21×10^{1}	**4.22×10^{-1}**	2.90×10^{1}	5.55×10^{2}	3.35	3.56×10^{1}
F10.Step	AB	**0**	**0**	**0**	**0**	**0**	**0**	5.33×10^{-1}	**0**	**0**	7.67	3.00×10^{-1}	2.90×10^{1}
	SD	**0**	**0**	**0**	**0**	**0**	**0**	1.83	**0**	**0**	1.52	5.96×10^{-1}	4.42×10^{1}
F11.Styblinski-Tang	AB	**-1.17×10^{3}**	-1.17×10^{3}	-1.17×10^{3}	-1.17×10^{3}	-1.17×10^{3}	-1.01×10^{3}	-1.05×10^{3}	-9.70×10^{2}	**-1.17×10^{3}**	-7.41×10^{2}	-1.14×10^{3}	-1.02×10^{3}
	SD	**2.31×10^{-13}**	1.66×10^{-9}	1.19×10^{1}	2.14×10^{-7}	2.58	4.16×10^{1}	2.85×10^{1}	3.91×10^{1}	**7.74×10^{-12}**	2.94×10^{1}	2.07×10^{1}	3.97×10^{1}
F12.Trid	AB	-1.28×10^{3}	3.11×10^{5}	**-4.65×10^{3}**	-4.39×10^{3}	**-4.93×10^{3}**	-1.42×10^{3}	-2.96×10^{3}	-5.93×10^{2}	2.50×10^{2}	1.00×10^{5}	-4.39×10^{3}	-2.40×10^{3}
	SD	2.56×10^{2}	7.26×10^{4}	**2.39×10^{2}**	2.67×10^{2}	**5.45×10^{-10}**	2.03×10^{3}	1.55×10^{3}	2.38×10^{4}	2.91×10^{3}	3.21×10^{4}	5.13×10^{2}	9.90×10^{2}
F13.Vincent	AB	**-3.00×10^{1}**	-2.81×10^{1}	-3.00×10^{1}	-3.00×10^{1}	-3.00×10^{1}	-3.00×10^{1}	-2.98×10^{1}	**-3.00×10^{1}**	-3.00×10^{1}	2.68×10^{14}	-2.95×10^{1}	-2.85
	SD	**0**	2.98×10^{-1}	7.53×10^{-3}	6.24×10^{-3}	2.60×10^{-3}	3.90×10^{-15}	5.67×10^{-1}	**0**	1.02×10^{-3}	6.61×10^{13}	2.92×10^{-1}	2.76
F14.Zakharov	AB	**3.39×10^{-5}**	1.85×10^{2}	3.98×10^{2}	6.79×10^{-1}	**4.10×10^{-11}**	7.69	1.42×10^{1}	1.97×10^{2}	5.75×10^{1}	1.09×10^{3}	9.02×10^{-3}	6.44×10^{2}
	SD	**4.96×10^{-5}**	2.61×10^{1}	9.24×10^{1}	3.96×10^{-1}	**6.40×10^{-11}**	2.05×10^{1}	2.60×10^{1}	6.74×10^{1}	2.18×10^{1}	3.23×10^{2}	9.12×10^{-3}	7.82×10^{2}
F15.Rotated hyper-ellisoid	AB	**1.64×10^{-104}**	9.86×10^{-10}	2.81×10^{-7}	7.95×10^{-6}	**7.20×10^{-25}**	6.91×10^{-12}	6.73×10^{-16}	5.22×10^{-15}	8.65×10^{-11}	2.08×10^{1}	5.49×10^{-22}	2.75×10^{-13}
	SD	**9.00×10^{-104}**	4.33×10^{-10}	1.52×10^{-6}	6.18×10^{-6}	**1.15×10^{-24}**	1.53×10^{-11}	1.78×10^{-15}	3.37×10^{-15}	1.20×10^{-10}	4.57	4.10×10^{-22}	1.26×10^{-12}
F16.Schwefel 2	AB	**1.33×10^{-64}**	2.41×10^{-8}	2.29×10^{-6}	2.56×10^{-4}	**3.16×10^{-23}**	3.33×10^{2}	9.00×10^{3}	9.92×10^{-14}	2.45×10^{-9}	6.81×10^{2}	3.81×10^{-19}	2.88×10^{-12}
	SD	**6.84×10^{-64}**	1.22×10^{-8}	1.21×10^{-5}	2.97×10^{-4}	**4.56×10^{-23}**	1.83×10^{3}	3.29×10^{4}	9.32×10^{-14}	5.02×10^{-9}	1.30×10^{2}	3.86×10^{-19}	5.07×10^{-12}
F17.Sphere	AB	**1.32×10^{-144}**	4.79×10^{-13}	5.99×10^{-11}	3.47×10^{-9}	**4.69×10^{-28}**	2.42×10^{-15}	5.44×10^{-18}	1.01×10^{-19}	5.38×10^{-14}	1.07×10^{-2}	1.70×10^{-25}	4.96×10^{-17}
	SD	**7.25×10^{-144}**	2.28×10^{-13}	3.08×10^{-10}	2.50×10^{-9}	**8.88×10^{-28}**	4.59×10^{-15}	2.65×10^{-17}	4.99×10^{-20}	1.86×10^{-13}	3.56×10^{-3}	1.90×10^{-25}	9.41×10^{-17}
F18.Sum squares	AB	**2.47×10^{-88}**	2.38×10^{-11}	1.19×10^{-10}	1.57×10^{-7}	**2.73×10^{-26}**	2.46×10^{-13}	7.89×10^{-16}	8.83×10^{-18}	2.92×10^{-12}	4.69×10^{-1}	6.95×10^{-23}	2.36×10^{-15}
	SD	**1.35×10^{-87}**	1.12×10^{-11}	3.15×10^{-10}	1.16×10^{-7}	**5.34×10^{-26}**	7.01×10^{-13}	3.58×10^{-15}	5.82×10^{-18}	6.77×10^{-12}	1.17×10^{-1}	1.09×10^{-22}	3.91×10^{-15}
F19.Sum of different powers	AB	**1.13×10^{-246}**	8.30×10^{-38}	5.55×10^{-21}	6.38×10^{-33}	2.29×10^{-59}	1.51×10^{-40}	**2.19×10^{-69}**	2.73×10^{-10}	3.92×10^{-48}	5.07×10^{-4}	8.09×10^{-57}	6.36×10^{-33}
	SD	**0**	1.44×10^{-37}	1.81×10^{-20}	1.66×10^{-32}	5.00×10^{-59}	4.04×10^{-40}	**7.36×10^{-69}**	3.03×10^{-10}	1.94×10^{-47}	5.22×10^{-4}	4.42×10^{-56}	3.19×10^{-32}
F20.Composite 1	AB	**2.31×10^{-82}**	8.89×10^{-7}	8.88×10^{-2}	2.86×10^{-4}	2.49×10^{-12}	7.67×10^{-9}	5.40×10^{-6}	7.90×10^{3}	3.19×10^{-7}	6.71×10^{2}	**5.68×10^{-13}**	1.44×10^{-8}
	SD	**1.26×10^{-81}**	2.18×10^{-7}	3.65×10^{-1}	9.25×10^{-5}	4.17×10^{-12}	6.68×10^{-9}	1.26×10^{-5}	2.32×10^{4}	2.92×10^{-7}	8.53×10^{2}	**2.21×10^{-13}**	3.03×10^{-8}
F21.Composite 2	AB	2.90×10^{1}	2.90×10^{1}	2.93×10^{1}	2.90×10^{1}	**2.90×10^{1}**	5.47×10^{1}	9.51×10^{1}	**2.90×10^{1}**	2.90×10^{1}	2.07×10^{2}	2.91×10^{1}	1.20×10^{2}
	SD	4.25×10^{-5}	1.58×10^{-8}	2.13×10^{-1}	2.13×10^{-5}	**1.33×10^{-14}**	1.28×10^{1}	2.03×10^{1}	**2.26×10^{-14}**	7.46×10^{-10}	1.45×10^{1}	7.58×10^{-1}	1.94×10^{1}
F22.Composite 3	AB	3.20×10^{1}	3.41×10^{2}	1.36×10^{2}	4.16×10^{1}	**3.20×10^{1}**	4.62×10^{1}	3.21×10^{1}	**3.20×10^{1}**	7.27×10^{1}	2.10×10^{5}	3.20×10^{1}	3.83×10^{1}
	SD	**3.84×10^{-6}**	4.35×10^{1}	6.87×10^{1}	5.24	**1.46×10^{-14}**	6.25	2.07×10^{-1}	3.58×10^{-5}	1.29×10^{1}	1.26×10^{5}	2.72×10^{-4}	4.05
F23.Composite 4	AB	**2.90×10^{1}**	2.90×10^{1}	2.96×10^{1}	2.90×10^{1}	**2.90×10^{1}**	4.17×10^{1}	8.35×10^{1}	1.46×10^{2}	2.90×10^{1}	3.67×10^{2}	2.90×10^{1}	1.47×10^{2}
	SD	**0**	1.75×10^{-6}	8.24×10^{-1}	3.99×10^{-4}	**1.62×10^{-11}**	1.13×10^{1}	2.52×10^{1}	2.79×10^{2}	1.04×10^{-6}	6.44×10^{1}	1.82×10^{-10}	3.69×10^{1}
F24.Shifted Ackley	AB	**0**	8.36×10^{-6}	2.37	2.47×10^{-4}	**9.38×10^{-14}**	1.27×10^{-7}	1.06	4.34×10^{-8}	1.29×10^{-6}	2.16	1.07×10^{13}	2.16
	SD	**0**	1.61×10^{-5}	7.15×10^{-1}	9.04×10^{-5}	**3.97×10^{-14}**	9.69×10^{-8}	8.20×10^{-1}	1.04×10^{-7}	1.44×10^{-6}	3.63×10^{-1}	3.97×10^{-3}	9.96×10^{-1}
F25.Shifted Powell	AB	**0**	1.27×10^{1}	4.85×10^{-2}	1.32×10^{-3}	**6.43×10^{-10}**	1.54×10^{-3}	1.04×10^{-4}	1.33×10^{-2}	2.08×10^{-2}	5.05×10^{2}	1.43×10^{-4}	1.56×10^{-3}
	SD	**0**	5.59	2.64×10^{-1}	1.25×10^{-3}	**7.95×10^{-10}**	6.31×10^{-4}	5.86×10^{-5}	8.94×10^{-3}	1.03×10^{-2}	1.48×10^{2}	8.68×10^{-5}	5.60×10^{-4}
F26.Shifted Rastringin	AB	**0**	5.17×10^{1}	5.27	2.01	4.70×10^{-3}	4.73×10^{1}	5.45×10^{1}	1.59×10^{2}	1.72×10^{-3}	2.24×10^{2}	**0**	9.55×10^{1}
	SD	**0**	4.15	1.33	1.55	3.59×10^{-3}	1.55×10^{1}	1.58×10^{1}	1.24×10^{1}	3.00×10^{-3}	1.29×10^{1}	**0**	7.52×10^{1}
F27.Shifted Rosenbrock	AB	**8.01×10^{-5}**	3.32×10^{1}	2.61×10^{1}	1.79	9.69	3.43×10^{1}	2.92×10^{1}	9.82	4.09×10^{1}	2.12×10^{3}	**0**	3.24×10^{1}
	SD	**1.79×10^{-4}**	8.18	1.98×10^{1}	6.22	9.58×10^{-1}	2.10×10^{1}	3.15×10^{1}	3.74×10^{-1}	2.65×10^{1}	7.37×10^{2}	**0**	3.49×10^{1}
F28.Shifted Rotated hyper-ellisoid	AB	**0**	1.02×10^{-9}	1.16×10^{-8}	7.61×10^{-6}	**5.78×10^{-25}**	7.83×10^{-12}	5.35×10^{-15}	5.04×10^{-15}	3.91×10^{-11}	2.19×10^{1}	8.40×10^{-22}	1.66×10^{-13}
	SD	**0**	4.63×10^{-10}	4.24×10^{-8}	4.79×10^{-6}	**1.16×10^{-24}**	1.57×10^{-11}	2.65×10^{-14}	3.21×10^{-15}	5.50×10^{-11}	5.29	8.36×10^{-22}	6.10×10^{-13}
F29.Shifted Schwefel 2	AB	**0**	2.18×10^{-8}	2.15×10^{-4}	2.58×10^{-4}	**4.50×10^{-23}**	3.00×10^{3}	9.33×10^{3}	1.10×10^{-13}	1.56×10^{-9}	7.43×10^{2}	5.99×10^{-19}	6.93×10^{-12}
	SD	**0**	9.66×10^{-9}	1.10×10^{-3}	1.57×10^{-4}	**8.03×10^{-23}**	1.64×10^{4}	2.46×10^{4}	6.61×10^{-14}	1.93×10^{-9}	2.04×10^{2}	6.21×10^{-19}	1.38×10^{-11}
F30.Shifted Schwefel 22	AB	**0**	5.34×10^{-6}	1.71	3.18×10^{-3}	1.84×10^{-7}	5.05×10^{1}	7.00×10^{1}	6.48×10^{38}	2.77×10^{-6}	5.68×10^{25}	**8.52×10^{-11}**	1.86×10^{1}
	SD	**0**	1.32×10^{-6}	9.57×10^{-1}	1.31×10^{-3}	5.06×10^{-7}	1.32×10^{2}	8.77×10^{1}	3.55×10^{39}	2.72×10^{-6}	1.35×10^{26}	**7.37×10^{-11}**	5.78×10^{1}
F31.Shifted Sphere	AB	**0**	4.98×10^{-13}	3.09×10^{-12}	3.28×10^{-9}	**3.93×10^{-28}**	2.71×10^{-15}	1.52×10^{-17}	1.10×10^{-19}	2.91×10^{-14}	1.04×10^{-2}	1.43×10^{-25}	2.59×10^{-16}
	SD	**0**	1.96×10^{-13}	1.31×10^{-11}	1.88×10^{-9}	**1.11×10^{-27}**	6.97×10^{-15}	6.28×10^{-17}	5.44×10^{-20}	7.11×10^{-14}	2.25×10^{-3}	1.30×10^{-25}	9.15×10^{-16}
F32.Shifted Sum of different powers	AB	**0**	2.00×10^{-11}	5.83×10^{-11}	1.63×10^{-7}	8.98×10^{-27}	7.76×10^{-14}	6.62×10^{-17}	6.39×10^{-18}	1.62×10^{-12}	4.59×10^{-1}	**0**	2.38×10^{-15}
	SD	**0**	6.63×10^{-12}	1.47×10^{-10}	1.22×10^{-7}	8.18×10^{-27}	1.77×10^{-13}	2.46×10^{-16}	3.80×10^{-18}	2.40×10^{-12}	1.32×10^{-1}	**0**	7.46×10^{-15}

TABLE V

PERFORMANCE OF THE TAILORED MUTATION–CROSSOVER COMBINATIONS SELECTED BY E^2-CTS-DE ACROSS DIFFERENT BENCHMARK FUNCTIONS AND DIMENSIONAL SETTINGS.

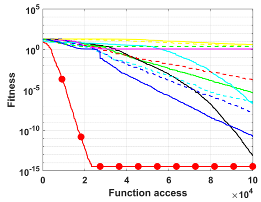
Function	Dims	Higher frequency		Lower frequency		BestSequence	Performance		
		Mutation op	Cross op	Mutation op	Cross op		Best	Avg	SD
F1	2	3,4	4	9	5	(4,4)	0	0	0
	10	6,9	4	8	5	(6,4)	0	0	0
	20	9,6	4	1	2	(6,4)	0	0	1.08×10^{-15}
	30	4	3	2	2	(9,3)	0	0	1.74×10^{-15}
F15	2	4	3,4	9	1	(4,3)	0	0	0
	10	6	3,4	1	2	(6,3)	0	3.50×10^{-215}	0
	20	9,6	4	1	2	(9,4)	1.16×10^{-198}	9.26×10^{-157}	5.07×10^{-15}
	30	4	4	1	2	(9,4)	1.49×10^{-176}	1.64×10^{-104}	9.00×10^{-104}
F20	2	4	4	9	2	(4,4)	0	0	0
	10	6	4	1	1	(6,4)	0	3.01×10^{-121}	1.65×10^{-120}
	20	9	4	1	2	(9,4)	2.23×10^{-104}	2.38×10^{-98}	1.15×10^{-97}
	30	6	4	1	1	(9,4)	1.22×10^{-93}	2.31×10^{-82}	1.26×10^{-81}
F24	2	3,6	4	8	5	(3,4)	0	0	0
	10	6	4,3	1	2	(6,4)	0	0	0
	20	9,6	4	1	2	(9,4)	0	0	0
	30	9	4	1	2	(6,4)	0	0	0

B. Statistical Significance

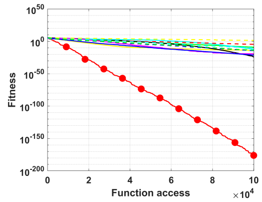
Table VI presents the results of the Friedman test for the evaluated algorithms. According to these results, E^2-CTS-DE is ranked first in the overall ordering. It is followed by LSHADE and KOA, which rank second and third, respectively, and also perform competitively across the benchmark set. Algorithms such as DSA, CMA-ES and BDE are placed in intermediate positions. In contrast, methods such as EHO and DMO achieve the highest ranking values, indicating relatively inferior performance compared with the other evaluated approaches. The obtained 2.74×10^{-33} confirms the presence of statistically significant differences among the considered algorithms, indicating that the observed variations are not due to random effects.

C. Operator-Selection Analysis

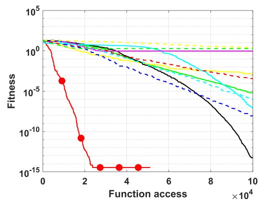
For this section, the objective is to analyze the behavior of the operator combinations selected by the proposed E^2-CTS-DE framework throughout the optimization process. Table V reports the usage frequencies of mutation and crossover operators, along with the best operator sequences identified across different benchmark functions and dimensional settings.



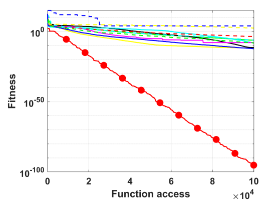
(a) Ackley.



(b) Rotated hyper-ellipsoid.



(c) Shifted Ackley.



(d) Composite 1.

(e)

Fig. 2. Best convergence profiles for selected benchmark functions used to compare E^2-CTS-DE and the reference algorithms.

TABLE VI
FRIEDMAN RANK TEST RESULTS ACROSS THE 32 BENCHMARK FUNCTIONS (LOWER AVERAGE RANK INDICATES BETTER PERFORMANCE).

Algorithm	Avg. rank	Order
E^2-CTS-DE	1.94	1
LSHADE	2.67	2
KOA	3.97	3
DSA	6.31	4
CMAES	6.52	5
BDE	7.00	6
PSO	7.33	7
TACPSO	7.42	8
DE	7.50	9
JADE	7.72	10
EHO	8.31	11
DMO	11.31	12
p-value		2.74×10^{-33}

From the observed results, the operator selection process does not uniformly favor all available options; instead, it converges toward combinations that have demonstrated superior performance during the iterative search. In the case of mutation, operators such as Gaussian-based mutation, Trigonometric mutation, and Hemostasis-based mutation appear recurrently among the most frequently selected, indicating that these strategies are particularly effective across different scenarios and dimensionalities, providing an appropriate balance between diversity and refinement. In contrast, operators such as DE/rand/1, DE/rand/2, and DE/current-to-rand/1, although present in some configurations, tend to become less relevant as dimensionality increases, appearing more often among the least selected options. This behavior suggests that while these mutations may promote exploration in the early stages, the adaptive selection scheme progressively prioritizes more consistent operators as the search progresses. Among crossover operators, the Directional crossover stands out as the most common choice, appearing consistently in both the highest-frequency combinations and the best-performing sequences, highlighting its ability to guide the recombination process effectively. Complementarily, the Continuous crossover emerges in specific cases, particularly in higher-dimensional settings. In contrast, Binomial, Exponential, and SBX show a more limited presence, mainly concentrated among the least frequent combinations. Overall, the patterns observed in Table V indicate that E^2-CTS-DE can identify and reinforce a reduced subset of robust mutation–crossover combinations, promoting more exploratory strategies when diversity is required and strengthening exploitation once promising regions of the search space are identified.

V. Conclusion

This paper proposes a learning-based Differential Evolution framework that integrates chaotic population initialization with an adaptive operator selection scheme guided by a Multi-Armed Bandit model and Thompson Sampling. Unlike traditional DE variants that rely on fixed or manually tuned operators, the proposed approach dynamically selects mutation–crossover combinations at the individual level, enabling a flexible search process driven by performance feedback. The experimental evaluation, conducted on a diverse benchmark set comprising multimodal, unimodal, composite, and shifted functions, demonstrates that E^2-CTS-DE achieves robust and competitive performance across different search landscapes. In particular, the algorithm exhibits clear advantages on multimodal and shifted functions, where effective exploration is critical, highlighting the benefits of combining diversity induced by chaotic maps with a learning-based operator selection mechanism. As the evolutionary process progresses, the proposed scheme gradually reinforces more exploitative strategies, promoting stable convergence. The analysis of operator usage frequency confirms that the Thompson Sampling mechanism can learn and reinforce suitable operator combinations based on problem characteristics and dimensionality, avoiding uniform selection and favoring configurations that consistently improve overall performance. This adaptive behavior naturally emerges from success–failure feedback, without requiring explicit rules or prior assumptions about operator effectiveness. Regarding mutation, operators such as Gaussian-based mutation (4), Trigonometric mutation (6), and Hemostasis-based mutation (9) appear recurrently both among the most frequently selected combinations and within the best-performing sequences, indicating that these strategies provide a balanced trade-off between diversity and solution refinement. Concerning crossover operators, the Directional crossover (4) consistently stands out as the most frequently selected and the one most often associated with high-performing sequences, indicating its ability to guide the recombination process effectively. The Continuous crossover (3) appears more selectively, mainly in specific dimensional settings, acting as a complementary option when more flexible recombination is required.

Additionally, the results indicate that learning-based operator selection combined with chaotic initialization constitutes an effective strategy for enhancing the performance of Differential Evolution. The proposed framework demonstrates generalization, stability, and robustness across diverse search landscapes. As future work, extend the framework to alternative bandit formulations, incorporate additional adaptive components, and evaluate its performance on real-world problems.

References

[1] 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, 2010.

[2] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu, "Hyper-heuristics: A survey of the state of the art," *Journal of the Operational Research Society*, vol. 64, no. 12, pp. 1695–1724, 2013.

[3] E. Burke, T. Curtois, M. Hyde, G. Kendall, G. Ochoa, S. Petrovic, J. A. Vázquez-Rodríguez, and M. Gendreau, "Iterated local search vs. hyper-heuristics: Towards general-purpose search algorithms," in *IEEE congress on evolutionary computation*. IEEE, 2010, pp. 1–8.

[4] G. L. Pappa, G. Ochoa, M. R. Hyde, A. A. Freitas, J. Woodward, and J. Swan, "Contrasting meta-learning and hyper-heuristic research: the role of evolutionary algorithms," *Genetic Programming and Evolvable Machines*, vol. 15, no. 1, pp. 3–35, 2014.

[5] E. Özcan, B. Bilgin, and E. E. Korkmaz, "A comprehensive analysis of hyper-heuristics," *Intelligent data analysis*, vol. 12, no. 1, pp. 3–23, 2008.

[6] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2, pp. 235–256, 2002.

[7] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[8] H. Kolsi, N. Benhadj, T. Guesmi, I. Marouani, B. M. Alshammari, H. Alsaif, K. Alqunun, and R. Neji, "Optimal design of a pmsm for electric vehicle using chaotic particle swarm optimization," *IEEE Access*, 2024.

[9] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[10] S. Wang, Y. Li, H. Yang, and H. Liu, "Self-adaptive differential evolution algorithm with improved mutation strategy," *Soft Computing*, vol. 22, no. 10, pp. 3433–3447, 2018.

[11] Z. Tan, K. Li, and Y. Wang, "Differential evolution with adaptive mutation strategy based on fitness landscape analysis," *Information Sciences*, vol. 549, pp. 142–163, 2021.

[12] G. Villalón-Sepúlveda, D. Oliva, and D. Zaldivar, "Hemostasis algorithm: A novel bio-inspired metaheuristic for global optimization," *Knowledge-Based Systems*, vol. 111, p. 106513, 2020.

[13] T. Eltaeib and A. Mahmood, "Differential evolution: A survey and analysis," *Applied Sciences*, vol. 8, no. 10, p. 1945, 2018.

[14] F. Herrera, M. Lozano, and J. L. Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis," *Artificial Intelligence Review*, vol. 12, pp. 265–319, 1998.

[15] A. K. Das and D. K. Pratihar, "A directional crossover (dx) operator for real parameter optimization using genetic algorithm," *Applied Intelligence*, vol. 49, no. 5, pp. 1841–1865, 2019.

[16] K. Deb, K. Sindhya, and T. Okabe, "Self-adaptive simulated binary crossover for real-parameter optimization," in *Proceedings of the 9th annual conference on genetic and evolutionary computation*, 2007, pp. 1187–1194.

[17] D. Bouneffouf, I. Rish, and C. Aggarwal, "Survey on applications of multi-armed and contextual bandits," in *2020 IEEE congress on evolutionary computation (CEC)*. IEEE, 2020, pp. 1–8.

[18] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem," in *Conference on learning theory*. JMLR Workshop and Conference Proceedings, 2012, pp. 39–1.

[19] G. C. Duque-Gimenez, D. F. Zambrano-Gutierrez, M. Rodriguez-Nieto, J. L. Menchaca, J. M. Cruz-Duarte, D. G. Zárate-Triviño, J. G. Avina-Cervantes, and J. C. Ortiz-Bayliss, "Viscoelastic characterization of the human osteosarcoma cancer cell line mg-63 using a fractional-order zener model through automated algorithm design and configuration," *Scientific Reports*, vol. 15, no. 1, p. 31436, 2025.

[20] A. J. Nebro, M. López-Ibáñez, J. García-Nieto, and C. A. Coello Coello, "On the automatic design of multi-objective particle swarm optimizers: experimentation and analysis," *Swarm Intelligence*, pp. 1–35, 2023.

[21] D. F. Zambrano-Gutierrez, J. M. Cruz-Duarte, J. G. Avina-Cervantes, J. C. Ortiz-Bayliss, J. J. Yanez-Borjas, and I. Amaya, "Automatic design of metaheuristics for practical engineering applications," *IEEE Access*, vol. 11, pp. 7262–7276, 2023.

[22] M. López-Ibáñez, J. Dubois-Lacoste, L. P. Cáceres, M. Birattari, and T. Stützle, "The irace package: Iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.

[23] A. B. Ozer, "Cide: chaotically initialized differential evolution," *Expert Systems with Applications*, vol. 37, no. 6, pp. 4632–4641, 2010.

[24] W. Wong and C. I. Ming, "A review on metaheuristic algorithms: recent trends, benchmarking and applications," in *2019 7th International Conference on Smart Computing & Communications (ICSCC)*. IEEE, 2019, pp. 1–5.

[25] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the american statistical association*, vol. 32, no. 200, pp. 675–701, 1937.

[26] J. Zhang and A. C. Sanderson, "Jade: Self-adaptive differential evolution with fast and reliable convergence performance," in *2007 IEEE congress on evolutionary computation*. IEEE, 2007, pp. 2251–2258.

[27] P. Civicioglu and E. Besdok, "Bezier search differential evolution algorithm for numerical function optimization," *Expert Systems with Applications*, vol. 165, 2020.

[28] R. Tanabe and A. S. Fukunaga, "Improving the search performance of shade using linear population size reduction," in *2014 IEEE congress on evolutionary computation (CEC)*. IEEE, 2014, pp. 1658–1665.

[29] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[30] T. Ziyu and Z. Dingxue, "A modified particle swarm optimization with an adaptive acceleration coefficients," in *2009 Asia-Pacific Conference on Information Processing*, vol. 2. IEEE, 2009, pp. 330–332.

[31] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)," *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.

[32] A. G. Hussien, M. Amin, M. Wang, G. Liang, A. Alsanad, A. Gumaei, and H. Chen, "Crow search algorithm: theory, recent advances, and applications," *Ieee Access*, vol. 8, pp. 173 548–173 565, 2020.

[33] M. Abdel-Basset, R. Mohamed, S. A. A. Azeem, M. Jameel, and M. Abouhawwash, "Kepler optimization algorithm: A new metaheuristic algorithm inspired by kepler's laws of planetary motion," *Knowledge-based systems*, vol. 268, p. 110454, 2023.

[34] M. A. Al-Betar, M. A. Awadallah, M. S. Braik, S. Makhadmeh, and I. A. Doush, "Elk herd optimizer: a novel nature-inspired metaheuristic algorithm," *Artificial Intelligence Review*, vol. 57, no. 3, p. 48, 2024.

[35] G. Moustafa, A. M. El-Rifaie, I. H. Smaili, A. Ginidi, A. M. Shaheen, A. F. Youssef, and M. A. Tolba, "An enhanced dwarf mongoose optimization algorithm for solving engineering problems," *Mathematics*, vol. 11, no. 15, p. 3297, 2023.

[36] P. Civicioglu, "Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm," *Computers & Geosciences*, vol. 46, pp. 229–247, 2012.