Journal of Network Intelligence
Taiwan Ubiquitous Information

Volume **9**, Number **1**, February **2024**

Differential Evolution Strategy with Chebyshev Chaos Based Mutation

Amos Sitenda, Pei-Cheng Song, Shu-Chuan Chu*

College of Computer Science and Engineering
Shandong University of Science and Technology, Qingdao 266590, China
amossitenda@gmail.com, spc7299@gmail.com, scchu0803@gmail.com

Shi-Huang Chen

Department of Computer Science and Information Engineering
Shu-Te UniversityKaohsiung, Kaohsiung 824, Taiwan
shchen@stu.edu.tw

*Corresponding author: Shu-Chuan Chu

Received September 22, 2023, revised November 24, 2023, accepted December 29, 2023.

Abstract. *Differential Evolution (DE) is a potent stochastic evolutionary optimization algorithm garnering increasing research attention. Over the years, it has been found applicable in solving diverse real-world problems. DE employs mutation, crossover, and selection operators to guide populations toward optimal or nearly optimal solutions. However, the standard DE mutation strategies have found limitations in balancing exploration and exploitation effectively, thus prompting research into possible improvements. This study introduces a novel mutation strategy named Chebyshev Infused Chaos Mutation Strategy (CICMS). where a chaotic sequence partially guides the process of donor vector generation. Rigorous evaluations were conducted, comparing our modified DE against the standard DE and seven other metaheuristic algorithms, including Genetic Algorithm, Particle Swarm Optimization and Gravitational Search Algorithm. Experiments were performed using the challenging CEC 2014 benchmark functions, consisting of 30 objective functions. Results indicate substantial improvements in convergence speed and solution quality, highlighting the potential of our novel mutation strategy to enhance DE's practicality in addressing complex optimization problems. This research contributes valuable insights to the dynamic field of optimization algorithms with implications for a wide range of applications.*
Keywords: Differential Evolution, Evolutionary optimization algorithm, Mutation, Chaos theory

---

1. **Introduction.** Differential Evolution (DE) is a powerful stochastic optimization algorithm that has garnered widespread recognition for its effectiveness in solving complex optimization problems. It is a method grounded in population-based optimization, inspired by the principles of natural selection. DE operates by maintaining a population of candidate solutions and iteratively evolving these solutions to discover optimal or near-optimal solutions for a diverse array of applications.

The mutation operator within the DE algorithm plays a pivotal role in shaping its effectiveness. Research in the field of optimization has underscored the vital significance of mutation operators, revealing their profound impact on the algorithm's convergence, exploration, and exploitation capabilities. These studies have illuminated the critical role of the mutation strategy in influencing the algorithm's performance and adaptability to various problem domains.

One of the foundational mutation strategies in DE is based on the work of Storn and Price [1], published in 1997. This strategy, known as the standard DE mutation strategy (DE/rand/1), forms the backbone of the algorithm's mutation process. It involves the selection of three distinct candidate solutions from the population and the creation of a mutant vector through a linear combination of these individuals. The mutant vector, combined with the existing population, serves as a guiding force for Differential Evolution (DE) in exploring the solution space while upholding its robustness and simplicity. Additional early strategies, such as DE/best/1 [2] introduced in the same paper, DE/rand-to-best/1, DE/rand/2, and DE/current-to-rand/1, were discovered. Despite their merits, these strategies often encountered challenges in striking the right balance between exploration and exploitation, prompting research into potential enhancements. Scientists have explored strategies involving adaptation and self-adaptation techniques in mutation [3], hybridization with other algorithms or heuristics [4], and the utilization of parallel and distributed strategies [5], among others, to address inherent weaknesses in native mutation strategies. Our research is thus grounded in and motivated by ongoing efforts to enhance the mutation process of Differential Evolution, leading us to explore the avenue of hybridization with chaotic strategies.

DE relies on a selection mechanism to determine which candidate solutions from the current population will act as parents for the next generation. Typically, a straightforward stochastic selection strategy is employed, where individuals are chosen based on their fitness, ensuring that better solutions have a higher chance of being selected.

In the upcoming section, we'll delve into the earliest mutational strategies, exploring their inherent weaknesses. These limitations have spurred extensive research into innovative, chaotic mutation strategies. Section 3 elaborates on the operational framework and fundamental principles of our proposed mutation strategy. In Section 4, we present the experiments conducted, as well as their outcomes, in which our DE algorithm incorporating Chebyshev Chaos is benchmarked against seven other metaheuristic algorithms. Finally, Section 5 provides a comprehensive conclusion.

2. Related Work.

2.1. Native Mutation Strategies.

The mutation process within DE plays a pivotal role in fostering diversity among candidate solutions, contributing significantly to the algorithm's ability to explore the solution space. The mutation strategy involves perturbing a candidate solution by taking the difference between two randomly selected individuals from the current population and applying it to a third individual. This mutation operation introduces diversity and exploration into the algorithm. In the realm of optimization algorithms, DE exhibits various mutation strategies to balance exploration and exploitation. The basic DE/rand/1 strategy randomly selects three individuals X_{r1}, X_{r2}, X_{r3} from the population and computes the difference between X_{r2} and X_{r3}.

$$v_i = X_{r1} + F \times (X_{r2} - X_{r3}) \tag{1}$$

This strategy, discussed by Storn and Price in 1997, promotes exploration by preventing the algorithm from getting trapped in local minima. Yet, the introduction of a scaling factor F adds a layer of complexity; a higher F value broadens the reach of exploration while a lower value fine-tunes the search, thereby promoting exploitation. However, the method has its downsides, such as the risk of stagnation, computational complexity, and the necessity for careful parameter tuning.

On the other hand, the DE/best/1 strategy aims to expedite convergence by incorporating the best individual X_{best} from the population.

$$v_i = X_{\text{best}} + F \times (X_{r1} - X_{r2}) \tag{2}$$

This biases the search towards exploitation, as noted in the same seminal work by Storn and Price [2].

Further variations exist, such as DE/current-to-best/1, which combines the current individual X_i with the best and a random individual.

$$v_i = X_i + F \times (X_{\text{best}} - X_i) + F \times (X_{r1} - X_{r2}) \tag{3}$$

This strategy strives for a balance between exploration and exploitation, blending the strengths of the current best solution and random individuals. However, the mehtho is more prone to challenges in effectively leveraging the combined strengths of these diverse solutions. Additionally, the reliance on two distinct factors F may complicate parameter tuning.

Lastly, DE/current-to-rand/1 adds a twist by focusing more on exploration, incorporating more randomness into the mutation.

$$v_i = X_i + F \times (X_{r1} - X_i) + F \times (X_{r2} - X_i) \tag{4}$$

This strategy's heavy emphasis on randomness may lead to erratic behavior, potentially hindering convergence and slowing down the algorithm's progress towards finding optimal solutions.

2.2. **Chaotic Mutation Strategies.** The integration of chaotic strategies into DE algorithms has become a crucial area in evolutionary algorithm research, enhancing their performance, particularly in avoiding local optima and ensuring robust global search capabilities. Chaos theory, a branch of mathematics, explores the behavior of dynamical systems that are highly sensitive to initial conditions, characterized by deterministic yet unpredictable sequences known as chaotic sequences. These sequences are instrumental in dynamically adjusting algorithm parameters or guiding the entire search process, thereby improving the adaptability and robustness of the algorithm in complex search spaces.

Significant advancements in the integration of chaotic strategies into DE algorithms have been made by several researchers, each contributing unique insights and methods to enhance the algorithm's performance in various optimization challenges. These diverse approaches serve to dynamically adapt parameters, enhance local search mechanisms, maintain population diversity, and balance exploration and exploitation. Consequently, the overall performance and efficiency of DE algorithms are elevated, making them more capable of tackling various complex optimization problems. Guo et al.'s work on Self-Adaptive Chaos Differential Evolution (SACDE) presents a chaos mutation factor and dynamically changing weighting and crossover factors, demonstrating significant improvements in convergence speed and accuracy [6]. Similarly, Mandal et al.'s research on optimal energy management of microgrids using a hybrid optimization technique combining DE and chaos theory addresses the challenges of maximizing societal benefits while reducing environmental impacts [7]. Liang et al.'s study explores the Chaos Differential Evolution Optimization Algorithm's application in Support Vector Regression Machine, showcasing its effectiveness in continuous problem-solving [8], while Krömer et al.'s research delves into the intersection of randomness and chaos in Genetic Algorithms and Differential Evolution, highlighting the potential of chaotic sequences in various designs of Evolutionary Algorithms [9]. In the domain of structural reliability-based optimal design, Khodam et al. developed an improved DE and Chaos Control method, demonstrating its potential in environmental and resource management [10].

Furthermore, Gao et al.'s work on Chaotic Local Search-Based Differential Evolution Algorithms offers insights into the application of chaos theory in optimization [11]. Peng et al. focused on parameter estimation for chaotic systems using DE, underscoring the algorithm's simplicity and rapid convergence [12]. Liu et al.'s model for Vessel Trajectory Prediction based on AIS sensor data employs Adaptive Chaos Differential Evolution Support Vector Regression (ACDE-SVR), showing higher prediction accuracy and stability [13].

Fang et al. introduced a Multi-Objective Differential Evolution-Chaos Shuffled Frog Leaping Algorithm for water resources system optimization, outperforming traditional methods under various conditions [14]. Gao et al.'s novel inversion mechanism with functional extrema model via DE effectively identifies time-delayed fractional order chaos systems, demonstrating high precision and robustness [15]. Yuan et al.'s research on hydrothermal scheduling using chaotic hybrid DE addresses the challenges of self-adaptive parameter settings in DE and effectively handles constraints in the optimization process [16]. Coelho et al. applied chaotic sequences to DE for image contrast enhancement, enhancing the algorithm's performance [17]. Wang et al.'s hybrid DE algorithm based on chaos and generalized opposition-based learning (GOBL) showcases enhanced searching efficiency and quality [18]. Gao et al. carried out research on incommensurate and hyper fractional-order chaotic systems, offering a novel mechanism for parameter and fractional order inversion estimation [19]. Building upon similar innovative approaches in the field of computational intelligence, Chu et al. introduced the Symbiotic Organism Search (SOS) algorithm with a multi-group quantum-behavior communication scheme, primarily focused on applications in wireless sensor networks [20]. Moreover, Chu et al.'s work extends to the practical application of this algorithm in wireless sensor networks, particularly through the optimization of the DV-hop algorithm for node localization, achieving higher localization accuracy than some existing algorithms. The blend of theoretical innovation and practical application in Chu et al.'s research exemplifies the dynamic evolution of computational intelligence techniques and their significant impact on solving complex, real-world problems, echoing the foundational work of Gao et al. in advancing the field of chaotic systems and their applications in computational methodologies.

Pan et al.'s study on Binary Fish Migration Optimization (BFMO) for power systems highlights the importance of overcoming high-dimensional challenges in evolutionary algorithms [21]. Furthermore, Kong et al.'s work on energy-aware routing in WSNs using genetic algorithms [22] and Dao et al.'s Parallel Bat Algorithm for job shop scheduling problems [23] also contribute to this evolving field. The Parallel Binary Cat Swarm Optimization (PBCSO) by Pan et al. [24] further illustrates the diverse applications of advanced evolutionary algorithms. The Phasmatodea Population Evolution Algorithm by Song et al. [25] demonstrates the continual innovation in this field. The Five Phases Algorithm by Wang et al. [26] shows the broad scope of evolutionary algorithms in solving complex problems. Wu et al.'s work on a lightweight authentication protocol for IoHT emphasizes the growing need for security in smart medical services [27]. The research by Yue et al. on the Equilibrium Optimizer for emotion classification from English speech signals highlights the integration of evolutionary algorithms in language processing and emotion recognition [28]. Meng et al.'s development of the QUATRE algorithm, with an adaptation of the evolution matrix and selection operation, reflects advancements in numerical optimization techniques [29]. The study by Gan et al. on anomaly rule detection in sequence data showcases the application of these algorithms in data engineering and knowledge extraction [30]. Biswas et al.'s research on chaos control assisted single-loop multi-objective reliability-based design optimization using differential evolution represents a significant advancement in optimization. Their work, detailed in [31], introduces an innovative approach that addresses the computational challenges of traditional reliability-based design optimization methods. By integrating chaos control theory, the algorithm's efficiency in converging to reliable solutions in complex multi-objective scenarios is enhanced.

Fan et al.'s study on infrared electric image segmentation using Fuzzy Renyi Entropy and a Chaos Differential Evolution Algorithm, as presented in [32], demonstrates the effectiveness of evolutionary algorithms in image processing. Their method optimizes the segmentation threshold of inherently fuzzy infrared images through a chaos-driven approach, crucial for precise thermal infrared imagery segmentation.

Davendra et al.'s paper on scheduling the Lot-Streaming Flowshop scheduling problem with setup time using chaos-induced Enhanced Differential Evolution, cited in [33], showcases the versatility of evolutionary algorithms in operations research. They incorporate chaotic dynamics into the algorithm, enhancing its performance in complex scheduling problems and demonstrating the potential of integrating chaos theory into stochastic optimization methods.

Luo et al.'s work on an effective chaos-driven differential evolution for multi-objective unbalanced transportation problems considering fuel consumption, referenced in [34], addresses key aspects of sustainable logistics. Their study develops a mathematical model for fuel consumption and demonstrates superior performance in optimizing transportation systems, balancing economic and environmental considerations.

Ahadzadeh et al.'s enhancement of the Differential Evolution algorithm, discussed in [35], introduces a new mutation strategy and chaos local search to address limitations of the classical DE algorithm. Their two-step differential evolution (2sDE) approach significantly improves convergence speed and accuracy, proving effective in complex optimization problems.

Lastly, Senkerik et al.'s exploration of chaos-driven Differential Evolution on shifted benchmark function sets, detailed in [36], extends evolutionary algorithms' application to higher-dimensional problem spaces. Their research uses discrete dissipative chaotic systems as pseudo-random number generators, underlining the potential of natural chaotic dynamics in enhancing evolutionary algorithms' performance.

Each of these works contributes to the evolving narrative of evolutionary algorithms, extending their applicability and efficiency in addressing modern-day computational challenges.

3. **The proposed algorithm.** Given the standard DE mutation function,

$$\vec{V}_{i,G} = \vec{X}_{r1,G} + F \cdot (\vec{X}_{r2,G} - \vec{X}_{r3,G}) \tag{5}$$

Where: - G represents the current generation.
- i denotes the index of the individual being mutated.
- $X_{r1,G}$, $X_{r2,G}$, and $X_{r3,G}$ are three distinct vectors selected from the current population at generation G.
- Among these, $X_{r1,G}$ acts as the base vector during mutation.
- $\vec{V}_{i,G}$ is the generated donor vector.

Our proposed method introduces a specialized form of mutation in DE, termed as 'Partially Chaotic Mutation.' This innovative approach employs the Chebyshev Chaotic Function to augment the mutation process. Specifically, we generate a chaotic sequence that originates from the fittest individual in the initial generation. This sequence is sustained throughout the entire evolutionary process and serves as the source for the base vector in each iteration. The base vector thus becomes one of the three vectors essential for constructing the mutant vector, thereby facilitating a more adaptive and directed mutation operation.

3.1. **Chebyshev Chaotic Function.** One commonly used chaotic function is the Chebyshev chaotic function. It is defined mathematically as:

$$x_{n+1} = \cos(n \cdot \cos^{-1}(x_n))$$

Where x_n is the current state, and x_{n+1} is the next state in the sequence. The parameter n is a discrete time step.

3.2. **Principle of Chaos Generation.** The Chebyshev chaotic function is crucial in our DE algorithm for generating chaotic sequences, which are pivotal for the mutation strategy. This function transforms an initial state x_0 into a new state x_1, where the transformation is highly sensitive to the initial value. Even minor variations in x_0 lead to significant differences in x_1, making this property essential for the unpredictable evolution of the sequence.

The sequence progresses with each iteration n, displaying a deterministic yet seemingly random nature. This behavior is due to the inherent sensitivity of the Chebyshev function and the incorporation of time n as a multiplier in its equation, enhancing the function's unpredictability. Such characteristics are particularly beneficial in optimization algorithms like DE, where a balance between exploration and exploitation is necessary.

In our implementation, the Chebyshev chaotic map is defined as:

$$a_{i+1} = \cos(i \times \arccos(a_i)) \tag{6}$$

This equation generates the chaotic sequence crucial to our mutation strategy.

The sequence generation in our DE algorithm starts with randomly generated initial values for each particle in the population. These random values are crucial in creating a diverse and effective search process. The sequence then evolves according to:

$$x_1 = \text{Random Initial Value}, \tag{7}$$

$$x_{n+1} = \cos(n \times \arccos(x_n)). \tag{8}$$

Here, x_1 represents the initial value, and x_{n+1} denotes the value at the $n+1$-th iteration.

The chaotic sequence is visualized as:

$$\begin{bmatrix} x_2 & x_3 & \dots & x_{G_{\max}} \end{bmatrix}$$

Each value x_i in this sequence acts as a base vector at its respective iteration.

The mutation strategy in our DE algorithm, influenced by the chaotic sequence, is outlined as:

$$\vec{V}_{i,G} = \vec{X}_{\text{chaotic},G} + F \cdot (\vec{X}_{\text{rand1},G} - \vec{X}_{\text{rand2},G}) \tag{9}$$

Here, $\vec{X}_{\text{chaotic},G}$ is the base vector selected from the chaotic sequence, and $\vec{X}_{\text{rand1},G}$ and $\vec{X}_{\text{rand2},G}$ are the two randomly selected vectors from the current generation. This modified equation deviates from the standard DE approach by incorporating the chaotic sequence, which adds an additional layer of complexity and effectiveness to the mutation process.

4. **Experimental results.** In our comprehensive experimentation, we employed a population comprising 40 individuals and allowed it to evolve over 1,000 generations. This study entailed a comparative analysis with seven other renowned metaheuristic algorithms, namely: Particle Swarm Optimization (PSO) [37], Cat Swarm Optimization (CSO) [38], Genetic Algorithm (GA) [39], Gravitational Search Algorithm (GSA) [40], Bat Algorithm (BA) [41], Black Hole Algorithm (BH) [42], and the conventional Differential Evolution algorithm [1].

To ensure the robustness of our results, each algorithm was rigorously tested 51 times using the benchmark functions from the CEC 2014 suite [43], which comprises a total of 30 distinct objective functions. The experiment was run on a machine with an AMD Ryzen 5 6600H processor, operating at 3.30 GHz, accompanied by 16.0 GB of installed RAM (15.2 GB usable), and utilizing a 64-bit operating system, x64-based processor configuration.

Experimental results showed that the Chebyshev Infused Chaos Mutation Strategy (CICMS) competes fairly with these state-of-the-art algorithms, emerging among the top 3 in all experiments on unimodal, multimodal, and hybrid objective functions. The graphs below illustrate the convergence speed of the different algorithms. In graphs Figure 1 through Figure 4, our improved algorithm, CICMS, is referred to as DE_chebyshev_chaotic.

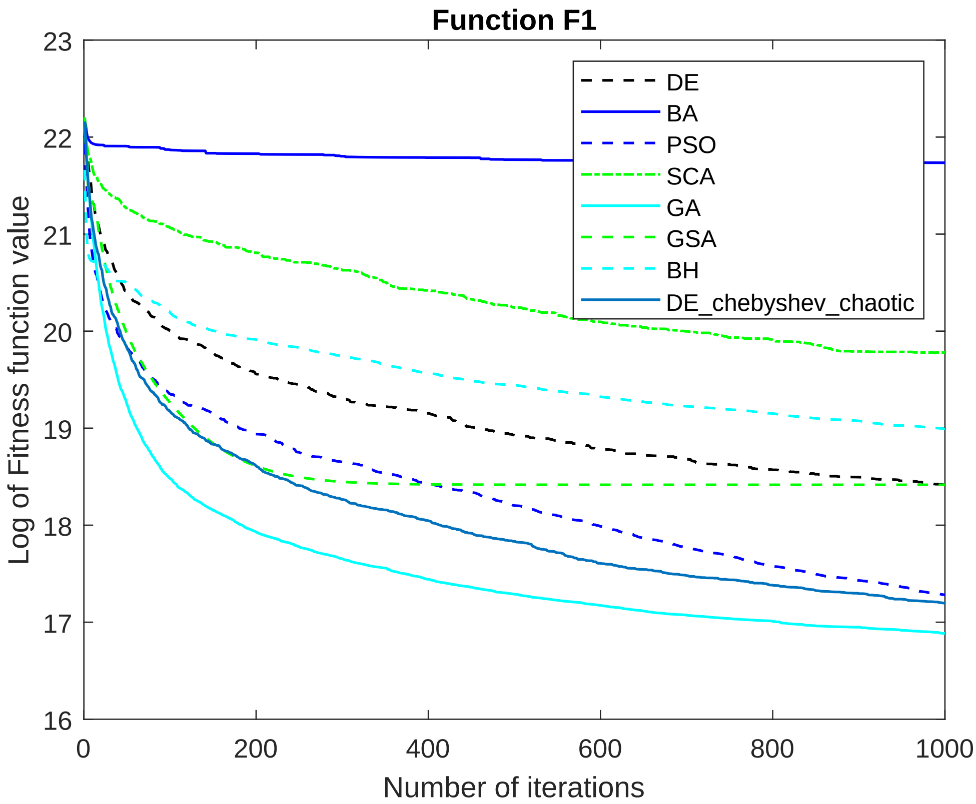

FIGURE 1. Unimodal Functions: Performance Comparison.

4.1. **Explanation for the graphs.** In testing unimodal function optimization(Figure 1), the CICMS algorithm demonstrated a superior ability to improve continuously without plateauing, suggesting it is robust in both exploring and exploiting the search space. Unlike the Bat Algorithm, which quickly finds a solution but then stalls—hinting at premature convergence—the DE Chebyshev algorithm showed no signs of such stagnation, an advantage in complex optimization. It outperformed PSO, SCA, and GSA algorithms in avoiding local optima and maintained progress, unlike DE and GA algorithms that did not sustain their initial advantage. The DE Chebyshev algorithm's consistent search for optimal solutions suggests a deeper strategy, beneficial for more complex problems and making it an attractive option for diverse optimization challenges.

For the optimization of the multimodal function F10(Figure 2), CICMS's graph indicates an ongoing, uninterrupted improvement, outstripping its counterparts. It avoids the early performance plateau that the Bat Algorithm (BA) encounters, which may suggest

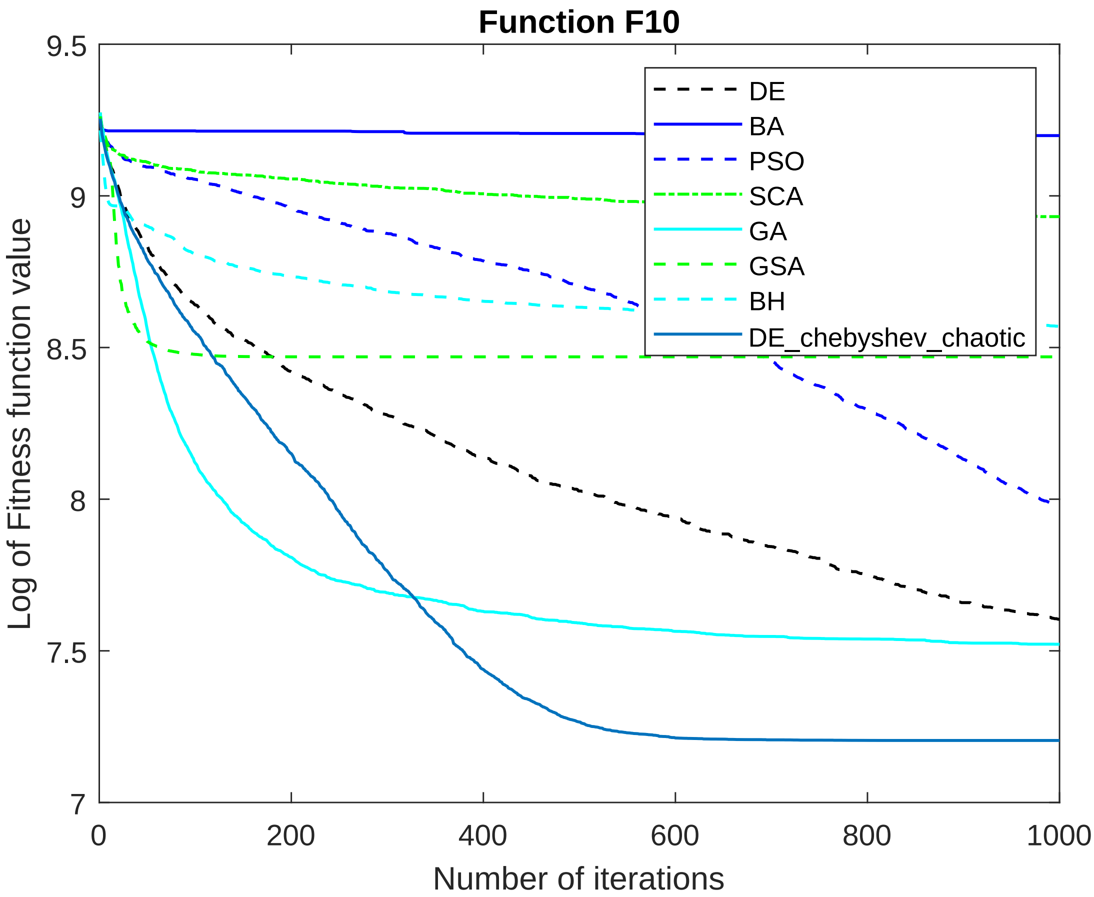

Figure 2. Multimodal Functions: Performance Comparison.

a quick settlement on suboptimal solutions. This enduring enhancement in performance indicates a comprehensive search strategy and robustness in solving complex optimization challenges, a notable advantage over PSO, SCA, and GSA algorithms that show a consistent, yet less dynamic, optimization path. The standard DE and GA algorithms, while starting strong, fail to maintain their initial momentum, contrasting with CICMS's persistent progress. Its steady advancement, devoid of stagnation, emphasizes its effectiveness in navigating the multifaceted landscapes of multimodal functions and underscores its potential in addressing more intricate optimization scenarios, affirming its suitability for a broad spectrum of optimization problems.

In optimizing hybrid function F21(Figure 3), CICMS outperforms others with its consistent improvement and no evident plateauing, suggesting robust optimization. It surpasses the Bat Algorithm, which stalls, indicating possible premature convergence. The algorithm also excels over PSO, SCA, and GSA by avoiding local optima and maintains a steady enhancement in solution quality, unlike the DE and GA algorithms which show diminishing returns. This sustained progress without stagnation demonstrates CICMS's proficiency in navigating the complexities of hybrid functions, proving its effectiveness for various optimization scenarios.

In the evaluation of composition function F27(Figure 4), our proposed algorithm maintains a notable downward trend in fitness value, indicative of effective optimization. It avoids the plateau observed in the Bat Algorithm (BA), implying a refined capability to escape premature convergence. In contrast to PSO, SCA, and GSA, CICMS's persistent decrease in the log of fitness function value suggests superior management of the complexities inherent in composition functions. It shows an advantage over the standard DE and

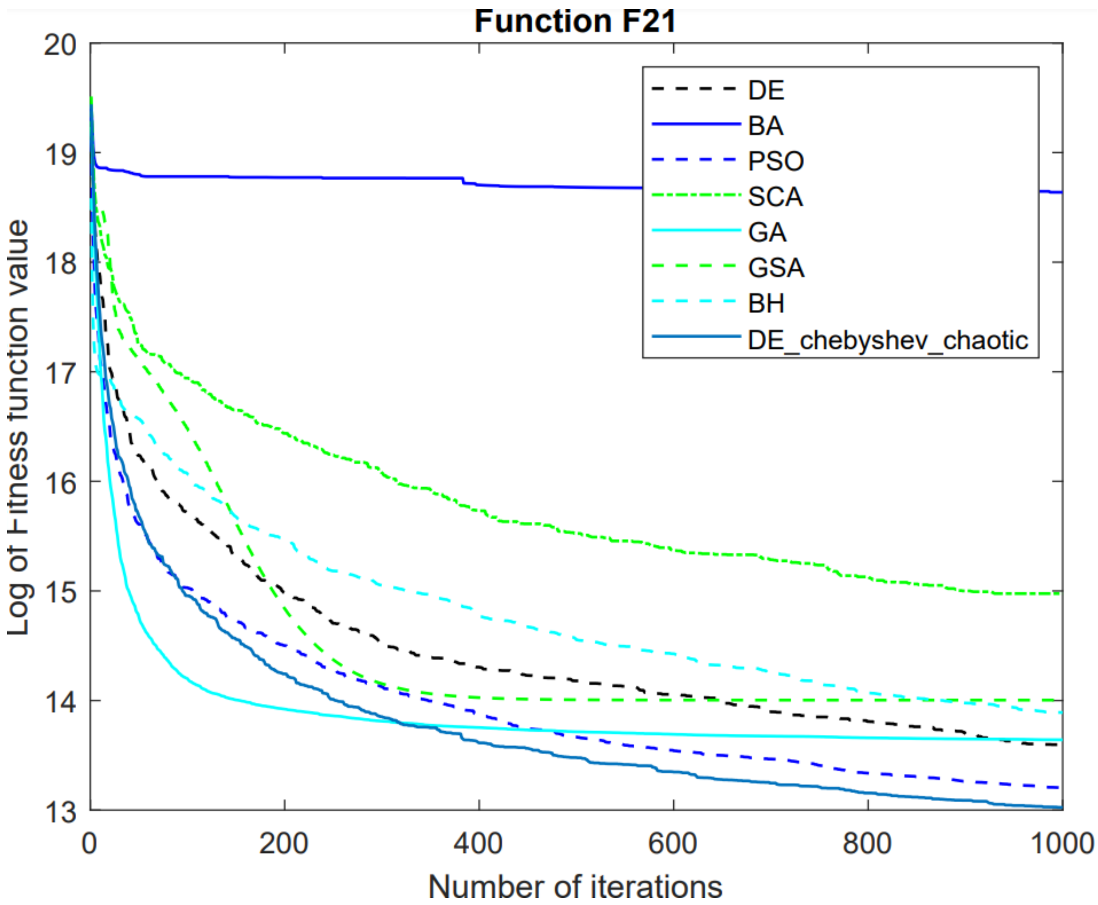

FIGURE 3. Hybrid Functions: Performance Comparison.

GA, which fail to sustain their initial rate of improvement. This graph depicts CICMS's adaptability and robustness, key in tackling the intricacies of composition functions, and affirms its utility across a diverse range of optimization problems.

4.2. **Comparing the mean, best value, and standard deviation.** For the unimodal functions (Functions 1 through 3 in tables 1 and 2), let's consider the Ellipsoidal Function where CICMS has a mean of $2.94E+7$, which is significantly better compared to the other algorithms like PSO ($3.2E+7$) and GA ($2.15E+7$). The standard deviation for CICMS is $1.22E+7$, indicating a variability that is less than BH ($4.2E+7$) and SCA ($1.13E+8$), suggesting more consistent performance. Moreover, CICMS's best value is $1.02E+7$, which is better than that of GA, GSA, and closely competes with PSO.

For multimodal functions (Function 4 to 16 in tables 1 and 2), if we take Ackley's Function as an example, all algorithms including CICMS have achieved the lowest possible mean and best values with negligible standard deviation, indicating a tie in performance.

Considering hybrid functions (Functions 17 to 22 in tables 1 and 2), Function 17 shows CICMS with a mean of $2.64E+6$, which is better than BH ($5.73E+6$) and GSA ($4.66E+6$), but not as good as PSO ($1.79E+6$). The standard deviation for CICMS is lower than that of BH and GSA, suggesting more reliability. The best value achieved by CICMS is also commendable at $4.48E+5$, surpassing BH and SCA.

For the composition functions (Functions 23 to 30 in tables 1 and tables 2), let's look at Function 27. Here, CICMS shows a mean of $3.2E+3$, which is competitive with other algorithms like BH ($3.27E+3$) and GSA ($4.5E+3$), indicating its efficacy. The standard deviation is low ($1.04E+2$), and the best value ($3.05E+3$) is better than that of BH and

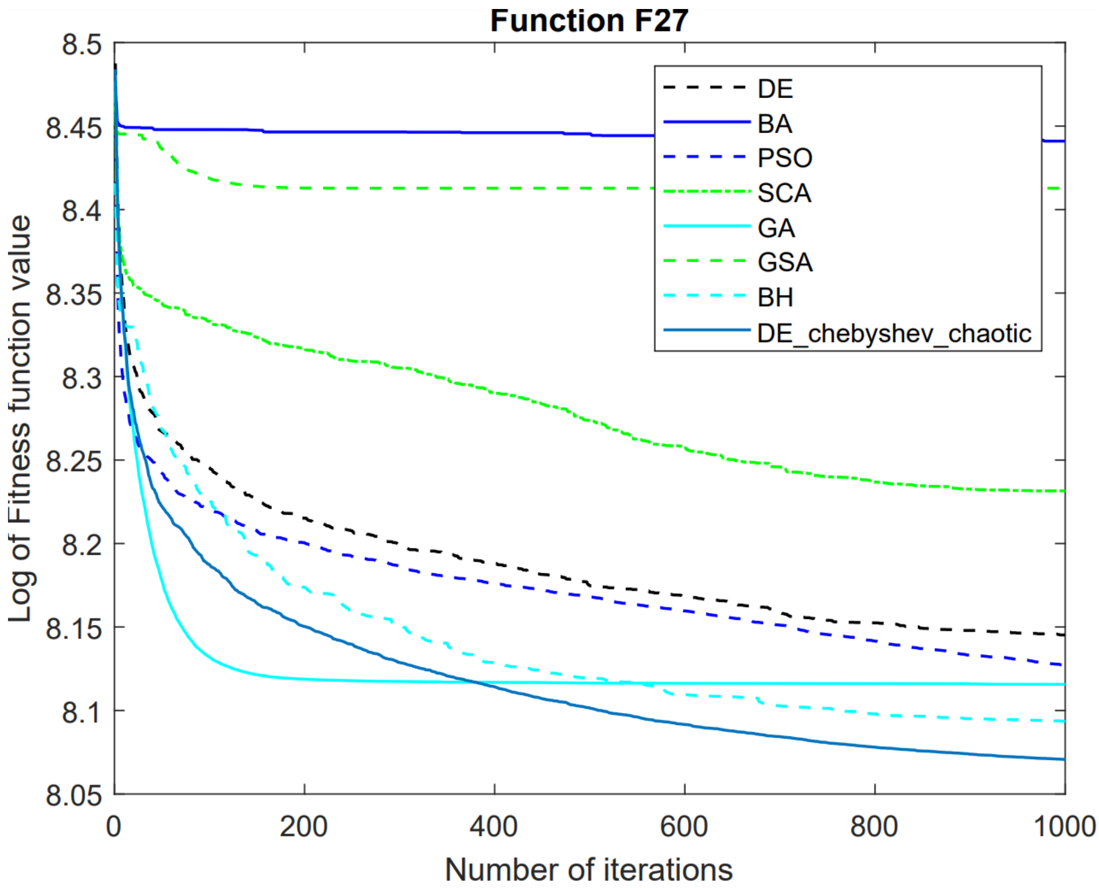

FIGURE 4. Composition Functions: Efficiency Analysis.

closely matches that of DE and GA, reinforcing its robust performance in composition functions.

TABLE 1. Performance Metrics For PSO, DE, GA and GSA

	PSO			DE			GA			GSA		
Func. Num	Mean	Best	Std	Mean	Best	Std	Mean	Best	Std	Mean	Best	Std
1	3.2E+7	5.03E+6	1.61E+7	9.95E+7	4.8E+7	2.49E+7	2.15E+7	5.12E+6	1.41E+7	9.95E+7	4.93E+7	2.73E+7
2	1.16E+7	1.32E+6	1.94E+7	1.1E+3	2.07E+2	2.4E+3	3.26E+7	7.93E+6	1.55E+7	4.62E+8	8.96E+3	4.47E+8
3	1.14E+4	1.5E+3	9.62E+3	2.36E+3	3.56E+2	1.94E+3	1.54E+4	8.5E+2	1.47E+4	8.27E+4	6.8E+4	7.63E+3
4	6.62E+2	5.01E+2	7.02E+1	5.44E+2	5.23E+2	1.36E+1	5.6E+2	4.89E+2	4.22E+1	8.63E+2	6.51E+2	1.74E+2
5	5.21E+2	5.21E+2	9.15E-2	5.21E+2	5.21E+2	6.04E-2	5.2E+2	5.2E+2	8.13E-2	5.2E+2	5.2E+2	6.71E-4
6	6.18E+2	6.12E+2	3.61E+0	6.29E+2	6.27E+2	1.31E+0	6.17E+2	6.09E+2	4.03E+0	6.26E+2	6.2E+2	2.93E+0
7	7.01E+2	7.01E+2	6.69E-2	7.E+2	7.E+2	1.45E-2	7.01E+2	7.01E+2	9.56E-2	7.11E+2	7.02E+2	8.44E+0
8	8.55E+2	8.24E+2	1.65E+1	8.52E+2	8.39E+2	5.2E+0	8.23E+2	8.15E+2	5.3E+0	9.43E+2	9.19E+2	1.32E+1
9	1.05E+3	9.64E+2	5.02E+1	1.07E+3	1.04E+3	1.41E+1	9.95E+2	9.46E+2	2.28E+1	1.07E+3	1.03E+3	1.88E+1
10	2.9E+3	2.04E+3	6.24E+2	2.01E+3	1.63E+3	1.71E+2	1.85E+3	1.3E+3	2.72E+2	4.77E+3	3.63E+3	5.14E+2
11	6.65E+3	3.21E+3	1.16E+3	7.05E+3	6.21E+3	3.09E+2	4.74E+3	3.42E+3	6.1E+2	5.55E+3	4.29E+3	5.88E+2
12	1.2E+3	1.2E+3	5.16E-1	1.2E+3	1.2E+3	2.03E-1	1.2E+3	1.2E+3	2.05E-1	1.2E+3	1.2E+3	5.29E-3
13	1.3E+3	1.3E+3	1.13E-1	1.3E+3	1.3E+3	5.81E-2	1.3E+3	1.3E+3	9.82E-2	1.3E+3	1.3E+3	4.42E-1
14	1.4E+3	1.4E+3	1.55E-1	1.4E+3	1.4E+3	8.25E-2	1.4E+3	1.4E+3	6.59E-2	1.4E+3	1.4E+3	4.84E+0
15	1.52E+3	1.51E+3	3.95E+0	1.52E+3	1.51E+3	1.35E+0	1.52E+3	1.51E+3	6.71E+0	1.56E+3	1.52E+3	2.63E+1
16	1.61E+3	1.61E+3	3.91E-1	1.61E+3	1.61E+3	1.91E-1	1.61E+3	1.61E+3	5.12E-1	1.61E+3	1.61E+3	2.54E-1
17	1.79E+6	4.41E+5	8.73E+5	4.46E+6	1.64E+6	1.5E+6	3.54E+6	5.35E+5	2.31E+6	4.66E+6	1.61E+6	2.61E+6
18	2.81E+5	1.9E+3	1.45E+6	2.45E+5	3.88E+4	1.67E+5	4.2E+3	1.85E+3	3.11E+3	2.44E+3	2.01E+3	5.22E+2
19	1.92E+3	1.91E+3	1.85E+1	1.91E+3	1.91E+3	1.92E+0	1.93E+3	1.91E+3	3.01E+1	2.04E+3	1.95E+3	4.45E+1
20	1.52E+4	3.81E+3	9.55E+3	1.09E+4	2.71E+3	4.19E+3	4.31E+4	3.98E+3	2.03E+4	1.49E+5	6.96E+4	6.83E+4
21	5.41E+5	2.18E+4	4.37E+5	8.03E+5	2.67E+5	3.32E+5	8.4E+5	1.25E+5	6.71E+5	1.21E+6	1.76E+5	1.65E+6
22	2.56E+3	2.22E+3	2.08E+2	2.6E+3	2.38E+3	1.14E+2	2.77E+3	2.36E+3	2.17E+2	3.28E+3	2.79E+3	2.77E+2
23	2.62E+3	2.62E+3	1.E+0	2.62E+3	2.62E+3	1.37E-3	2.62E+3	2.62E+3	4.34E-1	2.58E+3	2.5E+3	7.6E+1
24	2.64E+3	2.63E+3	5.17E+0	2.63E+3	2.63E+3	1.42E+0	2.63E+3	2.63E+3	5.4E+0	2.61E+3	2.6E+3	8.82E+0
25	2.71E+3	2.71E+3	3.43E+0	2.72E+3	2.71E+3	3.33E+0	2.72E+3	2.71E+3	5.01E+0	2.7E+3	2.7E+3	3.68E+0
26	2.74E+3	2.7E+3	6.81E+1	2.7E+3	2.7E+3	5.64E-2	2.73E+3	2.7E+3	4.51E+1	2.79E+3	2.72E+3	1.88E+1
27	3.39E+3	3.11E+3	1.81E+2	3.45E+3	3.19E+3	1.29E+2	3.35E+3	3.1E+3	1.55E+2	4.5E+3	3.03E+3	4.41E+2
28	4.52E+3	3.95E+3	5.27E+2	3.72E+3	3.67E+3	2.67E+1	3.97E+3	3.6E+3	3.34E+2	5.75E+3	4.24E+3	7.41E+2
29	4.66E+6	3.88E+3	1.31E+7	8.32E+3	4.32E+3	8.63E+3	4.54E+3	3.82E+3	5.12E+2	1.4E+4	3.1E+3	7.41E+4
30	1.3E+4	7.02E+3	5.78E+3	1.22E+4	7.5E+3	2.24E+3	9.93E+3	5.51E+3	2.45E+3	1.87E+5	3.2E+3	8.73E+4

TABLE 2. Performance Metrics For BH, SCA, BA and CICMS

	BH			SCA			BA			CICMS		
Func. Num	Mean	Best	Std	Mean	Best	Std	Mean	Best	Std	Mean	Best	Std
1	1.77E+8	9.54E+7	4.2E+7	3.89E+8	2.03E+8	1.13E+8	2.75E+9	1.35E+9	8.98E+8	2.94E+7	1.02E+7	1.22E+7
2	6.7E+9	4.23E+9	1.28E+9	2.56E+10	1.57E+10	4.28E+9	1.13E+11	7.01E+10	1.94E+10	1.22E+3	2.E+2	1.63E+3
3	3.33E+4	1.68E+4	8.4E+3	5.46E+4	3.39E+4	1.07E+4	2.54E+5	1.46E+5	6.29E+4	1.7E+3	3.86E+2	9.74E+2
4	1.34E+3	9.75E+2	2.38E+2	2.05E+3	1.32E+3	5.04E+2	2.76E+4	1.16E+4	8.81E+3	5.01E+2	4.13E+2	3.21E+1
5	5.21E+2	5.2E+2	1.32E-1	5.21E+2	5.21E+2	5.29E-2	5.21E+2	5.21E+2	8.44E-2	5.21E+2	5.21E+2	6.68E-2
6	6.35E+2	6.25E+2	3.6E+0	6.37E+2	6.3E+2	2.25E+0	6.48E+2	6.42E+2	2.07E+0	6.14E+2	6.04E+2	5.2E+0
7	7.83E+2	7.4E+2	2.07E+1	9.03E+2	8.39E+2	3.46E+1	1.58E+3	1.21E+3	1.95E+2	7.E+2	7.E+2	9.81E-3
8	9.56E+2	8.95E+2	3.19E+1	1.07E+3	1.02E+3	2.11E+1	1.25E+3	1.13E+3	4.64E+1	8.14E+2	8.05E+2	5.43E+0
9	1.1E+3	1.01E+3	4.03E+1	1.21E+3	1.16E+3	2.22E+1	1.43E+3	1.32E+3	5.45E+1	1.05E+3	1.01E+3	1.28E+1
10	5.27E+3	4.01E+3	7.15E+2	7.57E+3	6.43E+3	4.64E+2	9.88E+3	8.41E+3	5.77E+2	1.35E+3	1.02E+3	2.1E+2
11	6.5E+3	5.25E+3	6.66E+2	8.56E+3	7.77E+3	3.48E+2	1.04E+4	9.46E+3	4.67E+2	6.7E+3	6.11E+3	2.7E+2
12	1.2E+3	1.2E+3	4.61E-1	1.2E+3	1.2E+3	3.24E-1	1.21E+3	1.2E+3	1.05E+0	1.2E+3	1.2E+3	2.17E-1
13	1.3E+3	1.3E+3	2.84E-1	1.3E+3	1.3E+3	3.64E-1	1.31E+3	1.31E+3	1.19E+0	1.3E+3	1.3E+3	8.47E-2
14	1.43E+3	1.42E+3	5.21E+0	1.46E+3	1.44E+3	1.31E+1	1.73E+3	1.61E+3	5.72E+1	1.4E+3	1.4E+3	1.92E-1
15	1.64E+3	1.56E+3	5.86E+1	1.43E+4	3.01E+3	1.14E+4	7.69E+6	1.33E+6	5.9E+6	1.51E+3	1.51E+3	1.24E+0
16	1.61E+3	1.61E+3	4.75E-1	1.61E+3	1.61E+3	2.97E-1	1.61E+3	1.61E+3	2.39E-1	1.61E+3	1.61E+3	2.59E-1
17	5.73E+6	1.5E+6	3.03E+6	1.25E+7	3.14E+6	6.1E+6	2.75E+8	6.42E+7	1.63E+8	2.64E+6	4.48E+5	1.63E+6
18	2.51E+6	2.14E+3	5.31E+6	3.4E+8	8.91E+7	1.63E+8	7.61E+9	2.14E+9	2.82E+9	1.18E+4	2.04E+3	1.67E+4
19	2.02E+3	1.94E+3	3.69E+1	2.02E+3	1.97E+3	2.39E+1	2.84E+3	2.25E+3	3.55E+2	1.91E+3	1.9E+3	1.97E+0
20	2.87E+4	1.16E+4	9.2E+3	3.37E+4	1.31E+4	1.46E+4	8.56E+6	1.45E+5	1.07E+7	8.06E+3	3.57E+3	3.81E+3
21	1.08E+6	9.4E+4	5.58E+5	3.19E+6	3.72E+5	2.13E+6	1.24E+8	1.38E+7	8.42E+7	4.52E+5	1.11E+5	2.68E+5
22	3.14E+3	2.71E+3	2.21E+2	3.19E+3	2.73E+3	1.68E+2	4.1E+4	4.12E+3	6.77E+4	2.51E+3	2.26E+3	1.43E+2
23	2.66E+3	2.64E+3	8.03E+0	2.7E+3	2.66E+3	2.08E+1	4.E+3	3.33E+3	4.83E+2	2.62E+3	2.62E+3	2.17E-1
24	2.64E+3	2.63E+3	7.08E+0	2.61E+3	2.6E+3	7.25E+0	2.84E+3	2.77E+3	4.4E+1	2.63E+3	2.62E+3	6.41E+0
25	2.73E+3	2.72E+3	8.54E+0	2.74E+3	2.7E+3	1.03E+1	2.86E+3	2.76E+3	5.84E+1	2.71E+3	2.71E+3	2.84E+0
26	2.7E+3	2.7E+3	3.41E-1	2.7E+3	2.7E+3	4.74E-1	2.79E+3	2.71E+3	1.07E+2	2.7E+3	2.7E+3	7.34E-2
27	3.27E+3	3.15E+3	1.23E+2	3.76E+3	3.22E+3	3.19E+2	4.63E+3	4.01E+3	2.66E+2	3.2E+3	3.05E+3	1.04E+2
28	7.19E+3	5.74E+3	5.93E+2	5.49E+3	4.64E+3	4.51E+2	9.29E+3	6.77E+3	1.42E+3	3.67E+3	3.56E+3	5.29E+1
29	1.93E+7	4.32E+4	1.64E+7	3.05E+7	9.76E+6	1.19E+7	1.15E+8	3.44E+5	1.34E+8	1.77E+5	4.1E+3	1.18E+6
30	1.64E+5	5.78E+4	7.23E+4	5.05E+5	1.62E+5	1.79E+5	8.62E+6	2.76E+6	5.11E+6	8.9E+3	4.06E+3	6.04E+3

4.3. **Time Analysis of DE_Chebyshev Chaotic.** The CICMS algorithm demonstrates a comprehensive search approach, (Table 3), reflecting in its computational time across various functions. It generally consumes more time compared to PSO and GA, suggesting a trade-off for its thorough search capabilities. In certain complex scenarios, CICMS excels, indicating its efficiency in navigating intricate problem spaces. This efficiency becomes evident as it outperforms algorithms like BH in specific functions. The algorithm's performance showcases its potential in effectively balancing depth of search with computational resources, making it a valuable tool for certain optimization problems where precision is paramount.

TABLE 3. Mean time taken by each algorithm

Func. Num	PSO	DE	GA	GSA	BH	CICMS	SCA	BA
1	2.17E+1	5.62E+1	7.81E+1	7.94E+1	2.95E+1	9.91E+1	3.54E+1	3.16E+1
2	3.8E+0	2.71E+1	3.92E+1	4.13E+1	5.57E+0	5.79E+1	1.26E+1	9.44E+0
3	3.81E+0	2.55E+1	3.79E+1	4.11E+1	4.37E+0	5.52E+1	1.24E+1	9.29E+0
4	3.89E+0	2.52E+1	3.82E+1	4.11E+1	5.5E+0	5.66E+1	1.24E+1	9.45E+0
5	6.24E+0	2.72E+1	4.08E+1	4.33E+1	9.5E+0	5.49E+1	1.5E+1	1.19E+1
6	2.41E+2	2.75E+2	2.93E+2	2.91E+2	6.43E+3	2.19E+3	2.56E+2	1.1E+4
7	7.62E+0	3.31E+1	4.92E+1	5.13E+1	1.06E+1	6.98E+1	1.78E+1	1.42E+1
8	8.06E+0	4.32E+1	6.53E+1	6.86E+1	1.21E+1	9.48E+1	2.21E+1	1.76E+1
9	1.04E+1	4.8E+1	7.14E+1	7.44E+1	1.52E+1	9.85E+1	2.54E+1	2.05E+1
10	1.21E+1	5.1E+1	7.33E+1	7.62E+1	1.82E+1	1.04E+2	2.75E+1	2.2E+1
11	1.4E+1	5.26E+1	7.48E+1	7.66E+1	2.07E+1	1.03E+2	2.95E+1	2.35E+1
12	9.11E+1	1.31E+2	1.57E+2	1.56E+2	1.35E+2	1.83E+2	1.07E+2	1.02E+2
13	6.81E+0	4.43E+1	6.8E+1	7.17E+1	1.E+1	9.37E+1	2.16E+1	1.67E+1
14	6.2E+0	4.4E+1	6.81E+1	7.08E+1	9.3E+0	9.46E+1	2.14E+1	1.66E+1
15	1.06E+1	4.95E+1	7.17E+1	7.41E+1	1.35E+1	1.01E+2	2.54E+1	2.05E+1
16	1.07E+1	4.78E+1	7.26E+1	7.55E+1	1.67E+1	9.82E+1	2.55E+1	2.08E+1
17	1.71E+1	5.59E+1	7.94E+1	8.08E+1	2.17E+1	1.08E+2	3.29E+1	2.77E+1
18	8.E+0	4.64E+1	7.E+1	7.22E+1	1.08E+1	9.75E+1	2.33E+1	1.81E+1
19	7.98E+1	1.21E+2	1.47E+2	1.46E+2	1.11E+2	1.76E+2	9.68E+1	9.22E+1
20	8.98E+0	4.72E+1	7.12E+1	7.37E+1	1.07E+1	9.91E+1	2.45E+1	1.93E+1
21	1.29E+1	5.2E+1	7.62E+1	7.76E+1	1.65E+1	1.05E+2	2.85E+1	2.33E+1
22	1.85E+1	5.48E+1	7.69E+1	7.75E+1	2.62E+1	1.02E+2	3.3E+1	2.81E+1
23	3.79E+1	6.73E+1	8.46E+1	8.58E+1	5.2E+1	1.09E+2	5.01E+1	4.62E+1
24	2.09E+1	6.02E+1	8.4E+1	8.45E+1	2.97E+1	1.15E+2	3.62E+1	3.14E+1
25	3.97E+1	7.79E+1	1.01E+2	1.01E+2	5.65E+1	1.28E+2	5.62E+1	5.08E+1
26	3.95E+2	4.44E+2	4.77E+2	4.67E+2	5.43E+2	5.01E+2	4.17E+2	4.11E+2
27	3.84E+2	4.26E+2	4.45E+2	4.32E+2	5.15E+2	4.85E+2	3.95E+2	3.89E+2
28	5.19E+1	9.48E+1	1.19E+2	1.16E+2	7.49E+1	1.51E+2	6.93E+1	6.36E+1
29	9.34E+1	1.33E+2	1.57E+2	1.53E+2	1.24E+2	1.85E+2	1.08E+2	1.03E+2
30	2.72E+1	5.93E+1	7.88E+1	7.81E+1	3.53E+1	1.02E+2	4.02E+1	3.57E+1

Table 4. Wilcoxon rank-sum test ranking of the 7 algorithms as compared to our proposed algorithm

Func. Num	PSO	DE	GA	GSA	BH	SCA	BA
1	5.119E-1	5.288E-18	2.937E-4	4.703E-18	3.304E-18	3.304E-18	3.304E-18
2	3.304E-18	9.84E-1	3.304E-18	3.304E-18	3.304E-18	3.304E-18	3.304E-18
3	3.049E-14	3.154E-1	8.077E-14	3.304E-18	3.304E-18	3.304E-18	3.304E-18
4	3.375E-17	3.782E-11	6.486E-11	3.304E-18	3.304E-18	3.304E-18	3.304E-18
5	1.634E-14	2.962E-2	4.181E-18	3.304E-18	2.106E-13	3.504E-18	3.304E-18
6	8.784E-5	3.304E-18	3.995E-4	4.49E-17	3.504E-18	3.304E-18	3.304E-18
7	3.077E-18	2.682E-4	3.077E-18	3.077E-18	3.077E-18	3.077E-18	3.077E-18
8	4.429E-18	3.299E-18	2.547E-12	3.299E-18	3.299E-18	3.299E-18	3.299E-18
9	1.427E-1	1.239E-12	7.073E-17	7.415E-11	8.817E-13	3.304E-18	3.304E-18
10	3.304E-18	1.343E-17	3.748E-14	3.304E-18	3.304E-18	3.304E-18	3.304E-18
11	4.375E-1	5.098E-8	3.304E-18	3.723E-15	4.759E-2	3.304E-18	3.304E-18
12	8.859E-11	8.461E-1	4.434E-18	3.304E-18	9.05E-3	3.304E-18	3.304E-18
13	3.374E-3	1.36E-5	6.668E-2	5.734E-2	3.304E-18	3.304E-18	3.304E-18
14	5.363E-7	2.059E-1	7.095E-6	1.221E-6	3.304E-18	3.304E-18	3.304E-18
15	1.05E-16	5.676E-13	9.686E-4	3.304E-18	3.304E-18	3.304E-18	3.304E-18
16	2.323E-4	1.956E-10	7.224E-5	3.304E-18	2.777E-8	1.391E-16	3.304E-18
17	2.628E-2	2.382E-7	4.466E-2	9.111E-6	7.36E-9	1.315E-16	3.304E-18
18	8.514E-1	1.005E-17	3.191E-6	2.749E-14	3.768E-8	3.304E-18	3.304E-18
19	2.711E-6	4.954E-12	2.61E-14	3.304E-18	3.304E-18	3.304E-18	3.304E-18
20	1.841E-5	9.286E-5	1.175E-16	3.304E-18	2.013E-17	1.343E-17	3.304E-18
21	7.178E-1	3.399E-7	7.432E-4	6.099E-5	3.292E-9	8.381E-17	3.304E-18
22	2.415E-1	2.385E-4	5.135E-9	4.987E-18	7.963E-18	4.987E-18	3.304E-18
23	3.504E-18	2.214E-13	3.946E-14	6.11E-1	3.304E-18	3.304E-18	3.304E-18
24	4.065E-8	3.095E-3	3.627E-1	1.794E-17	1.256E-4	6.315E-17	3.304E-18
25	1.719E-2	1.111E-16	2.482E-5	8.65E-16	8.441E-18	1.019E-15	3.304E-18
26	7.095E-6	6.603E-7	1.829E-1	3.304E-18	6.06E-6	3.304E-18	3.304E-18
27	1.765E-6	9.895E-14	3.332E-5	6.315E-17	1.955E-5	1.551E-14	3.304E-18
28	3.304E-18	4.553E-8	7.215E-12	3.304E-18	3.304E-18	3.304E-18	3.304E-18
29	6.768E-2	6.668E-2	9.714E-9	1.15E-11	8.441E-18	3.304E-18	4.181E-18
30	2.673E-8	1.708E-9	1.222E-4	6.315E-17	3.304E-18	3.304E-18	3.304E-18

5. **Conclusions.** This study introduces an innovative mutation strategy for DE that incorporates chaos theory, aiming to overcome the limitations of standard DE in balancing exploration and exploitation. Through extensive testing with the CEC 2014 benchmark functions, our enhanced DE has demonstrated significant improvements over the original algorithm and several other metaheuristic algorithms in terms of convergence speed and solution quality. These results suggest that our novel approach can make DE more effective for solving complex optimization problems in various real-world applications.

Acknowledgment. I extend my heartfelt gratitude to Pei-Cheng Song, Shu-Chuan Chu, Shi-Huang Chen, and Jia Zhao for their invaluable support throughout my research. Their insights and expertise were instrumental in shaping the trajectory of my study. I am also profoundly grateful to Shandong University of Science and Technology for providing the necessary resources and an encouraging academic environment that was crucial for my research. The collaborative effort and the intellectual generosity of these individuals and the institution have left a significant imprint on my work, and for that, I am deeply thankful.

REFERENCES

[1] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[2] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization.* Berlin: Springer, 2005.

[3] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[4] P. Kaelo and M. M. Ali, "Differential evolution algorithms using hybrid mutation," *Computational Optimization and Applications*, vol. 37, no. 2, pp. 231–246, June 2007. [Online]. Available: https://doi.org/10.1007/s10589-007-9014-3

[5] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *2013 IEEE Congress on Evolutionary Computation*, 2013, pp. 71–78.

[6] G. Zhenyu, C. Bo, Y. Min, and C. Binggang, "Self-adaptive chaos differential evolution," in *Advances in Natural Computation*, L. Jiao, L. Wang, X.-b. Gao, J. Liu, and F. Wu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 972–975.

[7] S. Mandal and K. K. Mandal, "Optimal energy management of microgrids under environmental constraints using chaos enhanced differential evolution," *Renewable Energy Focus*, vol. 34, pp. 129–141, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1755008420300181

[8] W. Liang, L. Zhang, and M. Wang, "The chaos differential evolution optimization algorithm and its application to support vector regression machine." *Journal of Software*, vol. 6, no. 7, pp. 1297–1304, 2011.

[9] P. Krömer, V. Snáel, and I. Zelinka, "Randomness and chaos in genetic algorithms and differential evolution," in *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, 2013, pp. 196–201.

[10] A. Khodam, P. Mesbahi, M. Shayanfar, and B. M. Ayyub, "Global decoupling for structural reliability-based optimal design using improved differential evolution and chaos control," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, vol. 7, no. 1, p. 04020052, 2021. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/AJRUA6.0001097

[11] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic local search-based differential evolution algorithms for optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3954–3967, 2021.

[12] B. Peng, B. Liu, F.-Y. Zhang, and L. Wang, "Differential evolution algorithm-based parameter estimation for chaotic systems," *Chaos, Solitons & Fractals*, vol. 39, no. 5, pp. 2110–2118, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077907004390

[13] J. Liu, G. Shi, and K. Zhu, "Vessel trajectory prediction model based on ais sensor data and adaptive chaos differential evolution support vector regression (acde-svr)," *Applied Sciences*, vol. 9, no. 15, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/15/2983

[14] G. Fang, Y. Guo, X. Wen, X. Fu, X. Lei, Y. Tian, and T. Wang, "Multi-objective differential evolution-chaos shuffled frog leaping algorithm for water resources system optimization," *Water Resources Management*, vol. 32, no. 12, pp. 3835–3852, 2018.

[15] F. Gao, X. jing Lee, F. xia Fei, H. qing Tong, Y. fang Deng, and H. ling Zhao, "Identification time-delayed fractional order chaos with functional extrema model via differential evolution," *Expert Systems with Applications*, vol. 41, no. 4, Part 2, pp. 1601–1608, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417413006787

[16] X. Yuan, B. Cao, B. Yang, and Y. Yuan, "Hydrothermal scheduling using chaotic hybrid differential evolution," *Energy Conversion and Management*, vol. 49, no. 12, pp. 3627–3633, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0196890408002719

[17] L. dos Santos Coelho, J. G. Sauer, and M. Rudek, "Differential evolution optimization combined with chaotic sequences for image contrast enhancement," *Chaos, Solitons & Fractals*, vol. 42, no. 1, pp. 522–529, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077909000149

[18] J. Wang, Z. Wu, and H. Wang, "Hybrid differential evolution algorithm with chaos and generalized opposition-based learning," in *Advances in Computation and Intelligence*, Z. Cai, C. Hu, Z. Kang, and Y. Liu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 103–111.

[19] F. Gao, F. xia Fei, X. jing Lee, H. qing Tong, Y. fang Deng, and H. ling Zhao, "Inversion mechanism with functional extrema model for identification incommensurate and hyper fractional chaos via differential evolution," *Expert Systems with Applications*, vol. 41, no. 4, Part 2, pp. 1915–1927, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417413007082

[20] S.-C. Chu, Z.-G. Du, and J.-S. Pan, "Symbiotic organism search algorithm with multi-group quantum-behavior communication scheme applied in wireless sensor networks," *Applied Sciences*, vol. 10, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/3/930

[21] J.-S. Pan, P. Hu, and S.-C. Chu, "Binary fish migration optimization for solving unit commitment," *Energy*, vol. 226, p. 120329, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544221005788

[22] L. Kong, J.-S. Pan, V. Snášel, P.-W. Tsai, and T.-W. Sung, "An energy-aware routing protocol for wireless sensor network based on genetic algorithm," *Telecommunication Systems*, vol. 67, no. 3, pp. 451–463, 2018.

[23] T.-K. Dao, T.-S. Pan, T.-T. Nguyen, and J.-S. Pan, "Parallel bat algorithm for optimizing makespan in job shop scheduling problems," *Journal of Intelligent Manufacturing*, vol. 29, no. 2, pp. 451–462, 2018.

[24] J.-S. Pan, X.-F. Ji, A. Liang, K.-C. Huang, and S.-C. Chu, "Parallel binary cat swarm optimization with communication strategies for traveling salesman problem," *Journal of Internet Technology*, vol. 22, no. 7, pp. 1621–1633, December 2021.

[25] P.-C. Song, S.-C. Chu, J.-S. Pan, and H. Yang, "Phasmatodea population evolution algorithm and its application in length-changeable incremental extreme learning machine," in *2020 2nd International Conference on Industrial Artificial Intelligence (IAI)*, 2020, pp. 1–5.

[26] X. Wang, S.-C. Chu, V. Snášel, H. A. Shehadeh, and J.-S. Pan, "Five phases algorithm: A novel meta-heuristic algorithm and its application on economic load dispatch problem," *Journal of Internet Technology*, vol. 24, no. 4, pp. 837–848, July 2023.

[27] T.-Y. Wu, L. Wang, and C.-M. Chen, "Enhancing the security: A lightweight authentication and key agreement protocol for smart medical services in the ioht," *Mathematics*, vol. 11, no. 17, 2023. [Online]. Available: https://www.mdpi.com/2227-7390/11/17/3701

[28] L. Yue, P. Hu, S.-C. Chu, and J.-S. Pan, "Equilibrium optimizer for emotion classification from english speech signals," *IEEE Access*, vol. 11, pp. 134 217–134 229, 2023.

[29] Z. Meng and J. Zhang, "Quatre-ems: Quatre algorithm with novel adaptation of evolution matrix and selection operation for numerical optimization," *Information Sciences*, vol. 651, p. 119714, 2023.

[30] W. Gan, L. Chen, S. Wan, J. Chen, and C.-M. Chen, "Anomaly rule detection in sequence data," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[31] R. Biswas and D. Sharma, "Chaos control assisted single-loop multi-objective reliability-based design optimization using differential evolution," *Swarm and Evolutionary Computation*, p. 101340, 2023.

[32] S. Fan and S. Yang, "Infrared electric image segmentation using fuzzy renyi entropy and chaos differential evolution algorithm," in *2011 International Conference on Future Computer Sciences and Application*. IEEE, 2011, pp. 220–223.

[33] D. Davendra, M. Bialic-Davendra, and R. Senkerik, "Scheduling the lot-streaming flowshop scheduling problem with setup time with the chaos-induced enhanced differential evolution," in *2013 IEEE Symposium on Differential Evolution (SDE)*. IEEE, 2013, pp. 119–126.

[34] R. Luo, S. Ji, and T. Ji, "An effective chaos-driven differential evolution for multi-objective unbalanced transportation problem considering fuel consumption," *Applied Soft Computing*, vol. 101, p. 107058, 2021.

[35] B. Ahadzadeh and M. B. Menhaj, "A modified differential evolution algorithm based on a new mutation strategy and chaos local search for optimization problems," in *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*. IEEE, 2014, pp. 468–473.

[36] R. Senkerik, M. Pluhacek, I. Zelinka, Z. K. Oplatkova, R. Vala, and R. Jasek, "Performance of chaos driven differential evolution on shifted benchmark functions set," in *International Joint Conference SOCO'13-CISIS'13-ICEUTE'13: Salamanca, Spain, September 11th-13th, 2013 Proceedings*. Springer, 2014, pp. 41–50.

[37] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[38] S. Chu, P. Tsai, and J. Pan, "Cat swarm optimization. pacific rim international conference on artificial intelligence," 2006.

[39] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[40] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Gsa: A gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[41] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies For Optimization (NICSO 2010)*. Springer, 2010, pp. 65–74.

[42] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Information Sciences*, vol. 222, pp. 175–184, 2013.

[43] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, vol. 635, no. 2, 2013.