symmetry

MDPI

Article

Novel Hybrid Crayfish Optimization Algorithm and Self-Adaptive Differential Evolution for Solving Complex Optimization Problems

Hussam N. Fakhouri [1], Abdelraouf Ishtaiwi [1], Sharif Naser Makhadmeh [1,2], Mohammed Azmi Al-Betar [2,*] and Mohannad Alkhalaileh [3]

[1] Data Science and Artificial Intelligence Department, Faculty of Information Technology, University of Petra, Amman 1196, Jordan; hussam.fakhouri@uop.edu.jo (H.N.F.); aishtaiwi@uop.edu.jo (A.I.); sharif.makhadmeh@uop.edu.jo (S.N.M.)

[2] Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman P.O. Box 346, United Arab Emirates

[3] College of Education, Humanities and Social Sciences, Al Ain University, Al-Ain P.O. Box 64141, United Arab Emirates; mohannad.alkhalaileh@aau.ac.ae

* Correspondence: m.albetar@ajman.ac.ae

Abstract: This study presents the Hybrid COASaDE Optimizer, a novel combination of the Crayfish Optimization Algorithm (COA) and Self-adaptive Differential Evolution (SaDE), designed to address complex optimization challenges and solve engineering design problems. The hybrid approach leverages COA's efficient exploration mechanisms, inspired by crayfish behaviour, with the symmetry of SaDE's adaptive exploitation capabilities, characterized by its dynamic parameter adjustment. The balance between these two phases represents a symmetrical relationship wherein both components contribute equally and complementary to the algorithm's overall performance. This symmetry in design enables the Hybrid COASaDE to maintain consistent and robust performance across a diverse range of optimization problems. Experimental evaluations were conducted using CEC2022 and CEC2017 benchmark functions, demonstrating COASaDE's superior performance compared to state-of-the-art optimization algorithms. The results and statistical analyses confirm the robustness and efficiency of the Hybrid COASaDE in finding optimal solutions. Furthermore, the applicability of the Hybrid COASaDE was validated through several engineering design problems, where COASaDE outperformed other optimizers in achieving the optimal solution.

Keywords: hybrid COASaDE; crayfish optimization algorithm; self-adaptive differential evolution; metaheuristic algorithms; optimization

check for updates

Citation: Fakhouri, H.N.; Ishtaiwi, A.; Makhadmeh, S.N.; Al-Betar, M.A.; Alkhalaileh, M. Novel Hybrid Crayfish Optimization Algorithm and Self-Adaptive Differential Evolution for Solving Complex Optimization Problems. *Symmetry* **2024**, *16*, 927. https://doi.org/10.3390/sym16070927

Academic Editor: Alexander Zaslavski

Received: 10 June 2024
Revised: 10 July 2024
Accepted: 11 July 2024
Published: 19 July 2024

1. Introduction

Optimization, a crucial concept in numerous scientific and engineering fields, aims to improve the efficiency and functionality of systems, designs, or decisions [1]. This interdisciplinary field merges principles from mathematics, computer science, operations research, and engineering to enhance the performance of complex systems [2]. Optimization endeavors to find the optimal values of an objective function within a specified domain, often navigating through competing objectives and constraints [3].

The mathematical backbone of optimization is founded on calculus and linear algebra, enabling the development of sophisticated techniques such as linear programming, quadratic programming, and dynamic programming [4,5]. These methodologies have undergone significant evolution, broadening the scope and applicability of optimization across various sectors, including economics, logistics, network design, and machine learning [6]. However, optimization is most fundamental in algorithm design and analysis, where it enhances efficiency and reduces computational complexity. This is especially critical for managing the large-scale data and complex computations prevalent in the digital era [7]. Similarly, in engineering, optimization plays a vital role in system design to maximize efficiency and minimize costs, with notable applications in aerospace for flight trajectory optimization and in electrical engineering for optimizing circuit designs [8,9].

Moreover, optimization has significantly integrated symmetry in its process, which provides a framework for balancing various algorithmic components to achieve optimal performance [10,11]. In the context of optimization algorithms, symmetry ensures a harmonious integration of different processes such as exploration and exploitation, allowing each phase to contribute equally and effectively to the search for optimal solutions [12,13]. This balanced approach prevents the algorithm from becoming biased towards either extensive searching (exploration) or intensive refining (exploitation), thereby maintaining consistent performance across diverse problem landscapes [14]. By leveraging symmetry, optimization algorithms can navigate complex high-dimensional spaces more efficiently, improving their ability to find global optima while avoiding local traps. Consequently, symmetry is integral to designing robust and versatile optimization algorithms capable of addressing a wide array of engineering and computational challenges [15].

Global optimization techniques that rely on statistical models of objective functions are tailored for tackling “expensive” and “black box” problems. In these scenarios, the limited available information is captured by a statistical model. To handle the high cost associated with such problems, the theory of rational decision-making under uncertainty is particularly appropriate. The resulting algorithms are essentially sequences of decisions made in the face of uncertainty, as informed by the statistical model [16].

Stochastic optimization algorithms are distinguished by their strategic use of randomness to tackle optimization challenges [17]. Unlike deterministic algorithms that follow a strict rule set, stochastic algorithms incorporate probabilistic elements to enhance solution space exploration, proving effective in complex high-dimensional problems with numerous local optima [18,19].

The concept of hybridization in optimization combines multiple optimization strategies to exploit the strengths of individual methods and mitigate their weaknesses, thereby enhancing overall effectiveness [20]. This paper proposes the hybridization of the Crayfish Optimization Algorithm (COA) with the Self-adaptive Differential Evolution (SaDE) technique [21]. COA, known for its global search capability, and SaDE, recognized for its adaptability in adjusting mutation and crossover rates, can be synergized to form a robust hybrid system. This hybrid approach is particularly advantageous in complex landscapes such as engineering design problems, where it provides a dynamic balance between exploration and exploitation, and is adept at handling non-differentiable or discontinuous objective functions in scenarios where traditional methods falter [22].

This hybrid model leverages the randomness of stochastic methods to circumvent the “curse of dimensionality”, in which the solution space grows exponentially with increasing dimensions, making exhaustive searches infeasible [23]. Furthermore, hybrid algorithms can offer more reliable convergence to global optima by navigating efficiently through multiple local optima, allowing them to adapt to various problem settings from machine learning and artificial intelligence to logistics and financial modeling [2].

Hybrid optimization not only consolidates the stochastic elements of methods such as Stochastic Gradient Descent (SGD), Genetic Algorithms (GAs) [24], Particle Swarm Optimization (PSO) [25], and Ant Colony Optimization (ACO) [26], it also introduces versatility in handling discrete and continuous optimization scenarios; discrete hybrid methods are suitable for combinatorial problems such as scheduling, while continuous hybrid strategies are ideal for continuous variable problems such as parameter tuning in machine learning models and optimization in control systems [22]. Thus, the integration of various optimization techniques into a hybrid framework holds the potential to revolutionize the efficiency and applicability of optimization solutions across a broad spectrum of disciplines.

2. Problem Statement

The Self-adaptive Differential Evolution (SaDE) algorithm, despite its powerful adaptive exploitation capabilities, often encounters challenges related to premature convergence. This issue arises due to its intensive local search mechanisms; while efficient at refining solutions, these can sometimes lead to the algorithm becoming trapped in local optima. Conversely, the Crayfish Optimization Algorithm (COA) excels in global search thanks to its explorative strategies inspired by the natural behavior of crayfish. However, COA's extensive search mechanisms can occasionally lack the precision required for fine-tuning solutions to their optimal values, particularly in complex high-dimensional landscapes.

The selection of SaDE and COA for hybridization is driven by the complementary strengths of these two algorithms. SaDE's capability to dynamically adjust its parameters during the optimization process makes it highly adaptable and effective at exploitation and refining solutions to a high degree of accuracy. Meanwhile, COA's robust exploration phase ensures a comprehensive search of the solution space that mitigates the risk of premature convergence and enhances the algorithm's ability to locate global optima.

The decision to hybridize SaDE and COA was based on a thorough analysis of their individual performance characteristics and the specific requirements of engineering design problems. Engineering optimization often involves navigating complex multimodal landscapes with numerous constraints and conflicting objectives. In such scenarios, relying solely on either exploration or exploitation is insufficient. SaDE's adaptive mechanisms provide a dynamic approach to parameter adjustment, which is critical for maintaining the algorithm's responsiveness to different stages of the optimization process. This adaptability ensures that the exploitation phase is finely tuned to the evolving landscape of the problem.

COA's exploration phase, inspired by the natural foraging behavior of crayfish, introduces a diverse set of candidate solutions into the search space. This diversity is essential for avoiding local optima and ensuring a broad search that can uncover high-quality solutions across the entire solution space. The hybrid COASaDE optimizer leverages the strengths of both algorithms, creating a balanced approach that enhances both exploration and exploitation. This balance is crucial for addressing the complex and varied challenges posed by engineering design problems while ensuring that the optimizer is both versatile and robust.

In light of the diverse and complex nature of engineering design problems, a hybrid approach that balances exploration and exploitation is essential. Many metaheuristic algorithms have been proposed over the years, each with its own unique strengths and weaknesses. However, the strategic combination of COA's explorative power and SaDE's adaptive exploitation creates a synergistic effect that enhances the overall performance of the optimization process. This hybrid model is particularly well-suited for engineering applications, where both global search capabilities and local refinement are crucial for efficiently finding high-quality solutions.

In addition, the hybridization of SaDE and COA is a strategic choice aimed at overcoming the limitations of each algorithm when used in isolation. By combining their strengths, the Hybrid COASaDE optimizer provides a powerful tool for solving complex engineering design problems, demonstrating superior performance and reliability across a wide range of benchmark tests and real-world applications.

2.1. Paper Contribution

The primary contributions of this paper are summarized as follows:

- **Novel hybrid optimization algorithm:** We propose the Hybrid COASaDE optimizer, a strategic integration of the Crayfish Optimization Algorithm (COA) and Self-adaptive Differential Evolution (SaDE). This hybridization leverages COA's explorative efficiency, characterized by its unique three-stage behaviour-based strategy, and SaDE's adaptive exploitation abilities, marked by dynamic parameter adjustment. The novelty of this approach lies in the balanced synergy between COA's broad search capabilities and SaDE's fine-tuning mechanisms, resulting in an optimizer that excels at both exploration and exploitation.

- **Enhanced balance between exploration and exploitation:** The Hybrid COASaDE algorithm effectively addresses the balance between exploration and exploitation, a common challenge in optimization. It begins with COA's exploration phase, which extensively probes the solution space to avoid premature convergence. As potential solutions are identified, the algorithm transitions to SaDE's adaptive mechanisms, which fine-tune these solutions through self-adjusting mutation and crossover processes. This seamless transition ensures that the algorithm can robustly navigate complex high-dimensional landscapes while maintaining a balanced and effective search process throughout the optimization.
- **Engineering applications:** The applicability and effectiveness of the Hybrid COASaDE optimizer are validated through several engineering design problems, including welded beam, pressure vessel, spring, speed reducer, cantilever, I-beam, and three-bar truss designs. In each case, Hybrid COASaDE achieves high-quality solutions with improved convergence speeds. This demonstrates the optimizer's versatility and robustness, making it a valuable tool for a wide range of engineering applications.

2.2. Paper Structure

The rest of this paper is organized as follows. The Introduction outlines the research problem, significance, and main contributions. The Related Work section reviews existing optimization algorithms, focusing on COA and SaDE. Next, overviews of COA and SaDE detail their respective strategies and mechanisms. The Proposed Hybrid COASaDE Optimizer section introduces the integration of COA and SaDE. The Mathematical Model of Hybrid COASaDE section provides detailed mathematical formulations. The Experimental Results section presents benchmark evaluations and comparisons. The Diagram Analysis section includes convergence curves, search history plots, and statistical analyses. The Engineering Problems section demonstrates the optimizer's application to engineering design problems. Finally, the Conclusion summarizes our findings and suggests future research directions.

3. Related Work

Optimization techniques have seen extensive evolution across various disciplines, significantly driven by developments in mathematics, computer science, operations research, and engineering [27]. These advancements have led to critical enhancements in linear programming, quadratic programming, and dynamic programming, facilitating complex problem-solving across such diverse fields as economics, logistics, and machine learning [3,5,28].

The authors of [29] introduced an enhanced version of the Quantum-inspired Differential Evolution global optimization (QDE) algorithm called EMMSIQDE. EMMSIQDE incorporates a novel differential mutation strategy to improve search and descent capabilities, and employs a multipopulation mutation evolution mechanism to maintain population diversity and independence. Additionally, a feasible solution space transformation strategy is utilized to optimize the solution mapping process. Comparative evaluations on multidimensional unimodal and multimodal functions demonstrated that EMMSIQDE outperforms DE, QDE, QGA, and MSIQDE in terms of optimization ability, scalability, efficiency, and stability.

In [30], the authors introduced WOASCALF, a novel hybrid algorithm combining the Whale Optimization Algorithm (WOA) and the Sine Cosine Algorithm (SCA) with Levy flight distribution to address optimization problems, in particular for high-dimensional problems. WOA excels in exploration but has weaknesses in exploitation, while SCA also faces exploitation issues. The hybrid WOASCALF algorithm leverages the strengths of both algorithms and to improve search agent movement through Levy flight. Evaluations using 23 benchmark functions and three real-world engineering problems showed that WOASCALF outperforms other algorithms in exploration ability.

An enhanced Particle Swarm Optimization (PSO) algorithm named TO-PSO, which uses a quasi-random sequence (QRS) for population initialization to improve the convergence rate and diversity, was proposed in [31]. By incorporating a low-discrepancy sequence, namely, the torus sequence, into PSO for swarm initialization, this approach aims to more effectively address optimization problems. Evaluations on fifteen well known unimodal and multimodal benchmark test problems demonstrated that TO-PSO significantly outperforms traditional PSO, Sobol Sequence-initialized PSO (SO-PSO), and Halton sequence-initialized PSO (HO-PSO) in terms of cost function value, convergence rate, and diversity.

The authors of [32] introduced the Whale Optimization with Seagull Algorithm (WSOA), a hybrid algorithm designed to address the computational accuracy and premature convergence issues of the Seagull Optimization Algorithm (SOA). By combining the contraction-surrounding mechanism of the WOA with the spiral attack behaviour of SOA and incorporating a Levy flight strategy, the WSOA enhances the balance between exploration and exploitation. Evaluations using 25 benchmark test functions and comparisons with seven well known metaheuristic algorithms demonstrated WSOA's superior performance. Additionally, four engineering examples validated its effectiveness and feasibility.

In [33], the authors proposed a Memory-based Hybrid Crow Search Algorithm (MHCSA) combining the Crow Search Algorithm (CSA) and PSO to address CSA's limitations in memory representation and balancing exploration and exploitation. By initializing the memory element with PSO's best solution (pbest) and enhancing CSA's individuals' positions using both the global best (gbest) and pbest of PSO, the MHCSA improves diversity and search abilities. Adaptive functions replace CSA's fixed flight length and awareness probability to better balance exploration and exploitation. When tested on seventy-three benchmark functions and seven engineering design problems, MHCSA demonstrated superior performance, while eliminating early convergence and improving both accuracy and stability, ranking first among CSA, PSO, and other robust methods.

In [34], the authors introduced MBFPA, a hybrid metaheuristic algorithm combining the Butterfly Optimization Algorithm (MVO) and Flower Pollination Algorithm (FPA) based on a mutualism mechanism. While effective, MVO often faces issues with local optimality and premature convergence. The MBFPA enhances exploration through FPA and improves exploitation via symbiosis organism search in the mutualism phase, thereby accelerating convergence. Meanwhile, its adaptive switching probability balances exploration and exploitation. When evaluated on 49 standard test functions, MBFPA outperformed six basic and five hybrid metaheuristic algorithms. Additionally, it successfully solved five classic engineering problems, demonstrating feasibility, competitiveness, and strong application prospects.

The COA, despite its simplicity and innovation, suffers from reduced search efficiency and susceptibility to local optima in later stages. To address these issues, a modified version called MCOA, was proposed in [35]. The MCOA incorporates an environmental renewal mechanism to guide crayfish to better environments based on water quality factors along with a learning strategy based on ghost antagonism to avoid local optima. Performance evaluations using the IEEE CEC2020 benchmark function, four constrained engineering problems, and feature selection problems demonstrated significant improvements. The MCOA showed a 55.23% improvement in average fitness value and a 10.85% increase in accuracy for feature selection problems. The combination of these enhancements substantially boosts MCOA's optimization performance, offering valuable advancements in solving complex spatial and practical problems.

A modified COA incorporating a Gaussian Distribution (GD) parameter to assess and enhance population and fitness diversity was proposed by the authors of [36]. Their proposed method aims to improve solution quality and robustness in complex optimization environments. The performance of the proposed method was validated by optimizing 27 unimodal and multimodal benchmark functions, showing that the proposed method is more effective and efficient than the traditional COA in fewer iterations.

All of these studies underscore the necessity for enhancing optimization algorithms in order to address their inherent limitations, with a particular focus on the Crayfish Optimization Algorithm (COA). These investigations highlight the critical need for improved search efficiency, preventing premature convergence, and bolstering exploration capabilities to achieve more robust and effective optimization performance. The modifications and hybrid approaches proposed in these studies demonstrate significant advancements in overcoming the shortcomings of traditional algorithms, paving the way for more efficient and reliable solutions to complex optimization problems.

3.1. Overview of COA

The Crayfish Optimization Algorithm (COA) [37] is an innovative metaheuristic algorithm inspired by the foraging behavior of crayfish in nature. At its core, COA is structured around three distinct behavioral stages, each mimicking a specific aspect of the crayfish's life cycle: the summer resort stage for exploration, the competition stage for resource allocation, and the foraging stage for exploitation. The summer resort stage symbolizes the search for a cooler habitat, guiding the algorithm to explore diverse regions of the solution space to prevent premature convergence. The competition stage, in which crayfish contest for resources, reflects the algorithm's ability to intensify the search in promising areas. Finally, the foraging stage, akin to crayfish seeking food, represents the fine-tuning of solutions in the algorithm. A unique feature of the COA is its temperature-based mechanism, which controls the transitions between these stages, enabling the algorithm to adaptively balance exploration and exploitation based on the problem's landscape [37].

3.2. Overview of Self-Adaptive Differential Evolution (SaDE)

Self-adaptive Differential Evolution (SaDE) [38] is a variant of the traditional Differential Evolution (DE) algorithm augmented with a self-adapting mechanism to dynamically adjust its strategy parameters during the optimization process [38]. This enhancement addresses a crucial challenge in DE, namely, the selection of appropriate control parameters such as the scaling factor and crossover rate, which significantly influences the algorithm's performance. SaDE introduces a learning strategy that allows it to adaptively modify these parameters based on the ongoing success rate of trial vectors in the population. By analyzing the performance feedback from previous iterations, SaDE can effectively tune its mutation and crossover strategies, thereby improving its ability to navigate through complex optimization landscapes [38]. This self-adaptive nature empowers SaDE to maintain a delicate balance between exploration and exploitation, thereby enhancing its efficiency and effectiveness in finding global optima, particularly in complex multimodal optimization scenarios. The adaptability of SaDE makes it a robust and versatile choice for a wide array of optimization problems, allowing it to serve as a foundation for the development of more sophisticated and efficient metaheuristic algorithms [39].

4. Hybrid COASaDE Optimizer

This section introduces the COASaDE Hybrid Optimizer, which combines the Crayfish Optimization Algorithm (COA) with Self-Adaptive Differential Evolution (SaDE). The hybridization approach in our study strategically integrates COA and SaDE to leverage their distinctive strengths and address complex optimization challenges effectively.

The integration process merges the explorative efficiency of COA, known for its unique three-stage behavior-based strategy, with the adaptive exploitation abilities of SaDE, characterized by dynamic parameter adjustment. The hybrid algorithm initiates with COA's exploration phase by extensively probing the solution space to prevent premature convergence. As potential solutions are identified, the algorithm transitions to employing SaDE's adaptive mechanisms, which fine-tune these solutions through self-adjusting mutation and crossover processes.

This methodology ensures a comprehensive exploration followed by an efficient exploitation phase, with each algorithm complementing the other's limitations. The primary challenge in this hybridization lies in achieving a seamless transition between the algorithms while maintaining a balance that allows the strengths of COA and SaDE to synergistically enhance the search process while mitigating their respective weaknesses.

4.1. COASaDE Mathematical Model

1. Population Initialization:

The algorithm initializes the population within the search space boundaries using uniformly distributed random numbers. This ensures diverse starting points for the search process, which increases the chance of finding the global optimum, as shown in Equation (1):

$$\mathbf{X}_i^{(0)} = \mathbf{lb} + (\mathbf{ub} - \mathbf{lb}) \cdot \mathbf{r}, \quad \mathbf{r} \sim \mathcal{U}(0,1) \tag{1}$$

where $\mathbf{X}_i^{(0)}$ is the i-th individual's position in the initial population, $\mathbf{lb}$ and $\mathbf{ub}$ are the lower and upper boundaries of the search space, and $\mathbf{r}$ is a vector of uniformly distributed random numbers.

2. Evaluation:

Each individual's fitness is evaluated using the objective function. This step determines the quality of the solutions, guiding the search process, as shown in Equation (2):

$$f_i^{(0)} = f(\mathbf{X}_i^{(0)}) \tag{2}$$

where $f(\cdot)$ is the objective function and $f_i^{(0)}$ is the fitness of the i-th individual.

3. Mutation Factor and Crossover Rate Initialization:

The algorithm initializes the mutation factor and crossover rate uniformly for all individuals. This provides a balanced exploration and exploitation capability initially, as shown in Equation (3):

$$F_i^{(0)} = 0.5, \quad CR_i^{(0)} = 0.7 \tag{3}$$

where $F_i^{(0)}$ is the mutation factor and $CR_i^{(0)}$ is the crossover rate for the i-th individual.

4. Adaptive Update:

The mutation factor and crossover rate are updated adaptively based on past success, enhancing the algorithm's ability to adapt to different optimization landscapes dynamically, as shown in Equations (4) and (5):

$$F_i^{(t)} = \max(0.1, \min(0.9, \mathcal{N}(F_i^{(t)}, 0.1))) \tag{4}$$

$$CR_i^{(t)} = \max(0.1, \min(0.9, \mathcal{N}(CR_i^{(t)}, 0.1))) \tag{5}$$

where $\mathcal{N}(\mu, \sigma)$ represents a normal distribution with mean μ and standard deviation σ.

5. COA Behavior:

The algorithm applies COA-specific behaviours such as foraging and competition based on temperature, which helps to diversify the search process and avoid local optima, as shown in Equations (6) and (7):

$$C^{(t)} = 2 - \frac{t}{T} \tag{6}$$

$$\mathbf{X}_i^{(t)} = \begin{cases} \mathbf{X}_i^{(t-1)} + C^{(t)} \cdot \mathbf{r} \cdot (\mathbf{xf}^{(t)} - \mathbf{X}_i^{(t-1)}), & \text{if temp} > 30 \\ \mathbf{X}_i^{(t-1)} - \mathbf{X}_z^{(t-1)} + \mathbf{xf}^{(t)}, & \text{otherwise} \end{cases} \tag{7}$$

where $C^{(t)}$ is the decreasing factor, $\mathbf{r} \sim \mathcal{U}(0,1)$ is a random vector, $\mathbf{xf}^{(t)}$ is the food position, and $\mathbf{X}_z$ is a randomly chosen individual's position.

6. SaDE Mutation and Crossover:

Differential Evolution's mutation and crossover mechanisms are applied to generate trial vectors, promoting solution diversity and improving the exploration capabilities of the algorithm, as shown in Equations (8) and (9):

$$\mathbf{V}_i^{(t)} = \mathbf{X}_{r1}^{(t)} + F_i^{(t)} \cdot (\mathbf{X}_{r2}^{(t)} - \mathbf{X}_{r3}^{(t)}) \tag{8}$$

$$\mathbf{U}_i^{(t)} = \begin{cases} V_{i,j}^{(t)}, & \text{if } \mathbf{r}_j \leq CR_i^{(t)} \text{ or } j = j_{\text{rand}} \\ X_{i,j}^{(t)}, & \text{otherwise} \end{cases} \tag{9}$$

where $r1, r2, r3$ are distinct indices different from i, $\mathbf{V}_i^{(t)}$ is the mutant vector, $\mathbf{U}_i^{(t)}$ is the trial vector, $\mathbf{r}_j \sim \mathcal{U}(0,1)$ is a random number, and j_{rand} is a randomly chosen index.

7. Selection:

The fitness of the trial vectors is evaluated and individuals are updated if the new solution is better. This step ensures that only improved solutions are carried forward, enhancing convergence towards the optimum, as shown in Equations (10) and (11):

$$f(\mathbf{U}_i^{(t)}) \leq f(\mathbf{X}_i^{(t)}) \implies \mathbf{X}_i^{(t+1)} = \mathbf{U}_i^{(t)}, \quad f_i^{(t+1)} = f(\mathbf{U}_i^{(t)}) \tag{10}$$

$$f(\mathbf{U}_i^{(t)}) > f(\mathbf{X}_i^{(t)}) \implies \mathbf{X}_i^{(t+1)} = \mathbf{X}_i^{(t)}, \quad f_i^{(t+1)} = f(\mathbf{X}_i^{(t)}) \tag{11}$$

where $\mathbf{X}_i^{(t+1)}$ is the updated position of the i-th individual and $f_i^{(t+1)}$ is its fitness.

8. Final Outputs:

The best solution found is returned as the final result along with the convergence curve indicating the algorithm's performance over time, as shown in Equations (12) and (13):

$$\mathbf{best\_position} = \mathbf{X}_{i^*}^{(T)}, \quad \text{where } i^* = \arg\min_i f_i^{(T)} \tag{12}$$

$$\text{best\_fun} = f(\mathbf{best\_position}) \tag{13}$$

where **best_position** is the best solution found, best_fun is its fitness, and cuve_f(t) is the best fitness at each iteration t.

The Hybrid Crayfish Optimization Algorithm with Self-Adaptive Differential Evolution (COASaDE) combines the Crayfish Optimization Algorithm's behavior with the evolutionary strategies of Differential Evolution (DE). The algorithm begins by initializing the population, mutation factor, and crossover rate. It then evaluates the initial population's fitness to determine the best solution. As the iterations proceed, when the iteration count exceeds a predefined stabilization period, the mutation factor and crossover rate are adaptively updated based on past success. The COA parameters, including the decreasing factor and temperature, are updated as well. Depending on the temperature, either the COA foraging/competition behaviour or DE mutation and crossover are performed. The algorithm evaluates new positions and updates the individuals when improvements are found. The global best position and fitness are updated at each iteration. The process continues until the stopping criterion is met, at which point the algorithm returns the best fitness and position along with the convergence information. The flowchart visualizes these steps, starting from initialization and moving through population evaluation, parameter updates, behavioural checks, mutation/crossover processes, evaluation of new positions, and updating global bests before concluding with the final results. Each decision point and process in the flowchart is linked to specific equations that define the algorithm's operations. Please see Algorithm 1.

Algorithm 1 Pseudocode for the Hybrid COASaDE algorithm.

```
Initialize parameters: population **X**, mutation factor **F**, crossover rate **CR**, and fitness values (see Equation (1)).
Evaluate initial population and determine the best fitness and position (see Equation (2)).
while stopping criterion not met do
    if iteration count > stabilization period then
        Adaptively update F and CR (see Equations (4) and (5)).
    end if
    Update COA parameters: $C$ and temperature (see Equation (6)).
    for each individual $i$ do
        if high temperature then
            Perform COA foraging or competition behavior (see Equation (7)).
        else
            Perform DE mutation and crossover (see Equations (8) and (9)).
        end if
        Evaluate new position and update if better.
    end for
    Update global best position and fitness.
end while
Return best fitness and position, and convergence information (see Equations (12) and (13)).
```

4.2. Exploration Phase of Hybrid COASaDE

The exploration phase of COASaDE is critical for navigating the search space and uncovering potential solutions. This phase leverages the exploration capabilities of COA combined with the adaptive mutation and crossover mechanisms of SaDE to effectively balance exploration and exploitation. The algorithm employs diverse strategies, including the foraging, summer resort, and competition stages, to facilitate varied movement patterns among candidate solutions while adapting to environmental cues for efficient search space exploration. Adaptive parameter adjustment plays a significant role in this phase by dynamically fine-tuning the mutation factors (F) and crossover rates (CR) for a subset of the population. This adjustment ensures that the algorithm adapts to changing conditions and maintains an effective balance between exploration and exploitation based on solution performance. Additionally, randomization and diversity maintenance are crucial, as is the introduction of stochasticity to prevent premature convergence to suboptimal solutions. By incorporating random factors such as temperature conditions and individual selection for parameter adaptation, the algorithm maintains diversity within the population, preventing it from becoming stuck in local optima and ensuring comprehensive exploration of the search space.

4.3. Exploitation Phase of Hybrid COASaDE

The exploitation phase of the hybrid Crayfish Optimization Algorithm with Self-Adaptive Differential Evolution (see Figure 1) focuses on refining candidate solutions to enhance their quality and convergence towards optimal solutions. This phase utilizes the adaptive mutation and crossover mechanisms of SaDE to efficiently exploit promising regions of the search space. The algorithm leverages the refined candidate solutions from the exploration phase and applies mutation and crossover operations to iteratively improve them. Boundary handling ensures that the solutions remain within the defined search space, preventing the generation of infeasible solutions and maintaining optimization integrity. Fitness evaluation and updating are pivotal, as newly generated solutions are evaluated using the objective function and superior solutions replace their parents in the population. The exploitation phase operates iteratively over multiple generations, refining candidate solutions based on evaluation of the objective function. Through continuous updates based on mutation, crossover, and fitness, COASaDE converges towards optimal or near-optimal solutions, effectively exploiting the promising regions identified during the exploration phase.

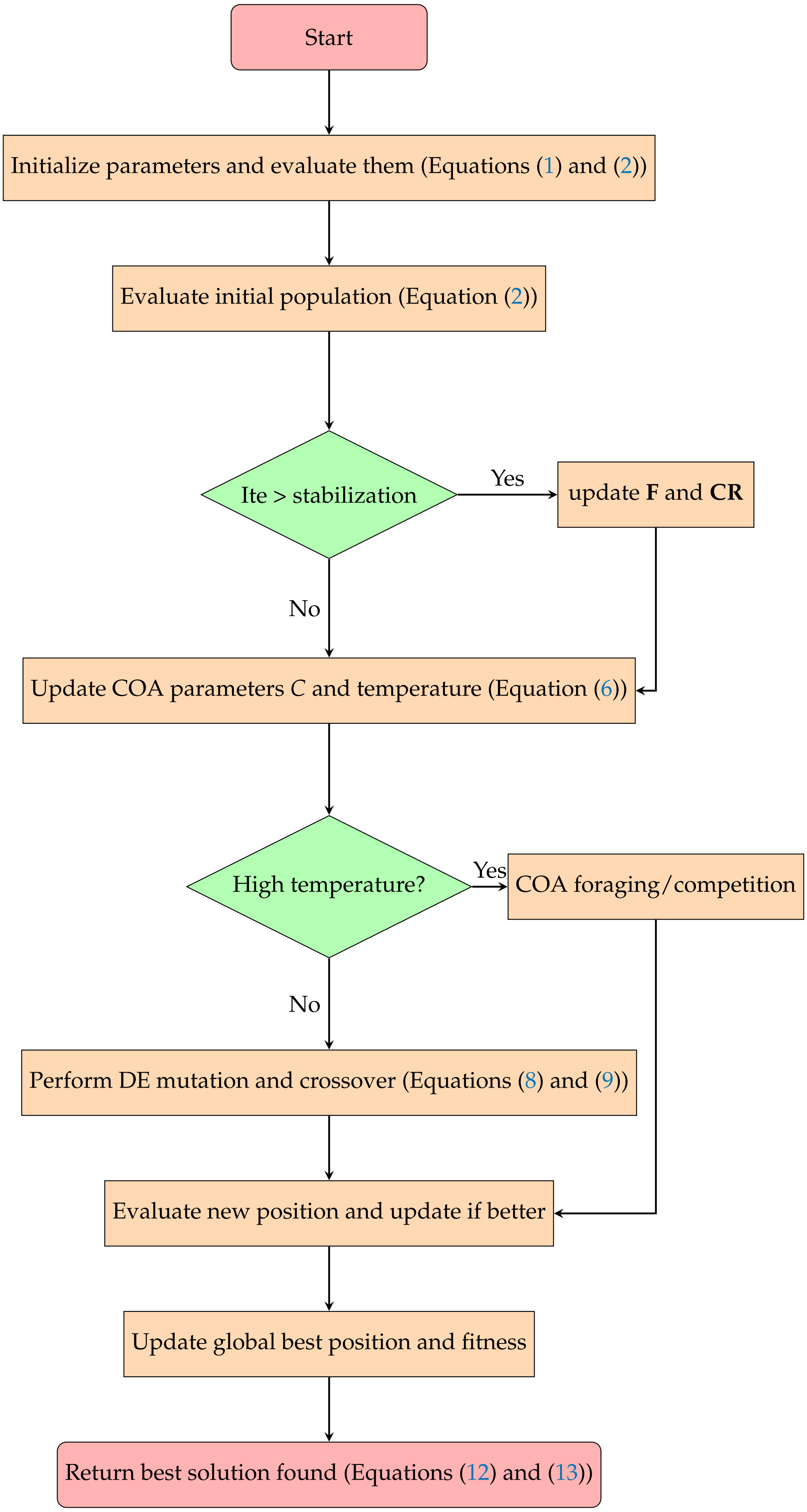
Start
Initialize parameters and evaluate them (Equations (1) and (2))
Evaluate initial population (Equation (2))
Ite > stabilization
Yes
update **F** and **CR**
No
Update COA parameters *C* and temperature (Equation (6))
High temperature?
Yes
COA foraging/competition
No
Perform DE mutation and crossover (Equations (8) and (9))
Evaluate new position and update if better
Update global best position and fitness
Return best solution found (Equations (12) and (13))

Figure 1. Flowchart of the proposed Hybrid COASaDE algorithm.

4.4. Solving Global Optimization Problems Using COASaDE

Global optimization problems involve finding the best possible solution from all feasible solutions, often in the presence of complex constraints and multiple local optima. These problems are characterized by their "black box" nature, where the objective function and constraints might not have explicit forms, making traditional analytical methods impractical [16]. Instead, heuristic and metaheuristic algorithms such as Differential Evolution, Genetic Algorithms, and Hybrid Optimization techniques are employed to efficiently navigate the search space. These algorithms iteratively explore and exploit the solution space, aiming to converge on the global optimum despite the challenges posed by the problem's landscape [40].

The general formulation of a global optimization problem can be expressed mathematically. This involves defining an objective function to be minimized or maximized and specifying the constraints that the solution must satisfy. The objective function, denoted as $f(\mathbf{x})$, and the constraints are illustrated in Equation (14):

$$\begin{aligned} &\text{Minimize (or Maximize):} \quad f(\mathbf{x}) \\ &\text{Subject to:} \\ &g_j(\mathbf{x}) \leq 0 \quad \text{for } j = 1, 2, \ldots, m \\ &h_i(\mathbf{x}) = 0 \quad \text{for } i = 1, 2, \ldots, p \end{aligned} \tag{14}$$

where:

$\mathbf{x} \in \mathbb{R}^n$ is the vector of decision variables, $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is the objective function, $g_j(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ for $j = 1, 2, \ldots, m$ are the inequality constraints, and $h_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, 2, \ldots, p$ are the equality constraints.

4.5. Symmetry in Hybrid COASaDE

The novel Hybrid COASaDE algorithm represents a fusion of mathematical theory and heuristic strategies, showcasing remarkable symmetry in its approach to optimization. Inspired by crayfish foraging behavior, COASaDE combines local and global search capabilities with heuristic rules derived from natural behaviors, effectively guiding the search process. The Differential Evolution (DE) framework, which underpins SaDE, uses mathematical models for differential mutation and crossover to explore the search space, while its self-adaptive mechanism adjusts control parameters dynamically based on the current optimization state. This hybrid optimizer balances exploration and exploitation by incorporating mathematical models and adaptive heuristics in order to use population-based search mechanisms to evolve candidate solutions. The synergy between COA's random and directed movements and SaDE's differential evolution framework enhances both global optimization and local search adaptability. This integration leverages the strengths of both algorithms, combining rigorous mathematical foundations with adaptive, nature-inspired heuristics to effectively solve complex optimization problems.

4.6. Computational Complexity of COASaDE

The computational complexity of COASaDE is influenced by several factors inherent to both Crayfish Optimization Algorithm (COA) and Self-Adaptive Differential Evolution (SaDE) that remain when combining them in a hybrid approach.

4.7. Computational Components

- **Initialization:**

$$O(N \cdot D) \tag{15}$$

The population X is initialized with N individuals (Equation (15)), where D is the dimensionality of the search space.

- **Fitness evaluation:**

$$O(N \cdot M) \tag{16}$$

The fitness of each individual in X is evaluated, where M is the complexity of the objective function.

- **Adaptive parameter updating:**

$$O(N) \tag{17}$$

The adaptive parameters F and CR for individuals are periodically updated, typically at every tenth iteration beyond $\frac{T}{10}$.

- **COA and SaDE operations:**

$$O(N \cdot D) \tag{18}$$

This controls COA behaviors (Equation (18)) and alternative COA behaviors (if condition $temp > 30$).

$$O(N \cdot D) \tag{19}$$

This controls SaDE mutation and crossover operations (Equation (19)).

- **Boundary conditions and fitness updating:**

$$O(N \cdot D) \tag{20}$$

This is used to apply the boundary conditions.

$$O(N \cdot M) \tag{21}$$

This is used to evaluate the fitness of mutated individuals.

- **Termination:**

$$O(T \cdot (N \cdot D + N \cdot M)) \tag{22}$$

The algorithm runs for T iterations, with T directly influencing the total computational complexity.

The computational complexity of COASaDE per iteration is primarily governed by the population size N, the dimensionality of the problem space D, the complexity of the objective function M, and the number of iterations T. The proposed hybrid approach aims to efficiently tackle complex optimization problems by combining adaptive parameter control and diverse search strategies from COA and SaDE.

5. Testing and Comparison

In our experimental testing and evaluation of COASaDE, we aimed to validate its efficacy and performance against a series of challenging benchmark functions. This section presents a comprehensive analysis of the results obtained from testing COASaDE on the CEC2017 and CEC2022 benchmark suites. The analysis involved comparing the performance of COASaDE with several state of art optimization algorithms. The comparative study focused on various metrics such as convergence speed, accuracy, robustness, and ability to escape local optima, providing a clear perspective on the strengths and potential limitations of the proposed COASaDE algorithm.

The selection of these algorithms was driven by several key considerations. First was the aim of representing a diverse range of optimization techniques. Algorithms such as GWO, PSO, MFO, MVO, SHIO, AOA, FOX, FVIM, and SCA embody different paradigms, from evolutionary and swarm intelligence-based methods to physics-inspired and hybrid approaches. This diversity allows researchers to conduct a comprehensive evaluation of COASaDE against established benchmarks to ensure a thorough exploration of its relative strengths and weaknesses across varied optimization scenarios.

Moreover, the choice of algorithms was guided by their status as benchmark standards within the optimization community. Algorithms such as WOA, GWO, and SCA have been extensively studied and cited, and as such can provide a reliable basis for comparison against newer or hybrid approaches like COASaDE. By benchmarking against well known algorithms, researchers can more effectively assess the performance of COASaDE, contributing to a deeper understanding of its capabilities and potential advancements in optimization research.

5.1. CEC2022 Benchmark Description

The CEC2022 benchmark suite represents a critical development in the field of optimization, focusing on single-objective and bound-constrained numerical optimization problems (see Figure 2). As a part of the evolutionary progression from previous competitions such as CEC'05 through CEC'21, which focused on real-parameter optimization, the CEC2022 suite introduces challenges based on substantial feedback from the CEC'20 suite. This feedback has been instrumental in refining the problem definitions and evaluation criteria in order to better assess the capabilities of modern optimization algorithms such as COASaDE.

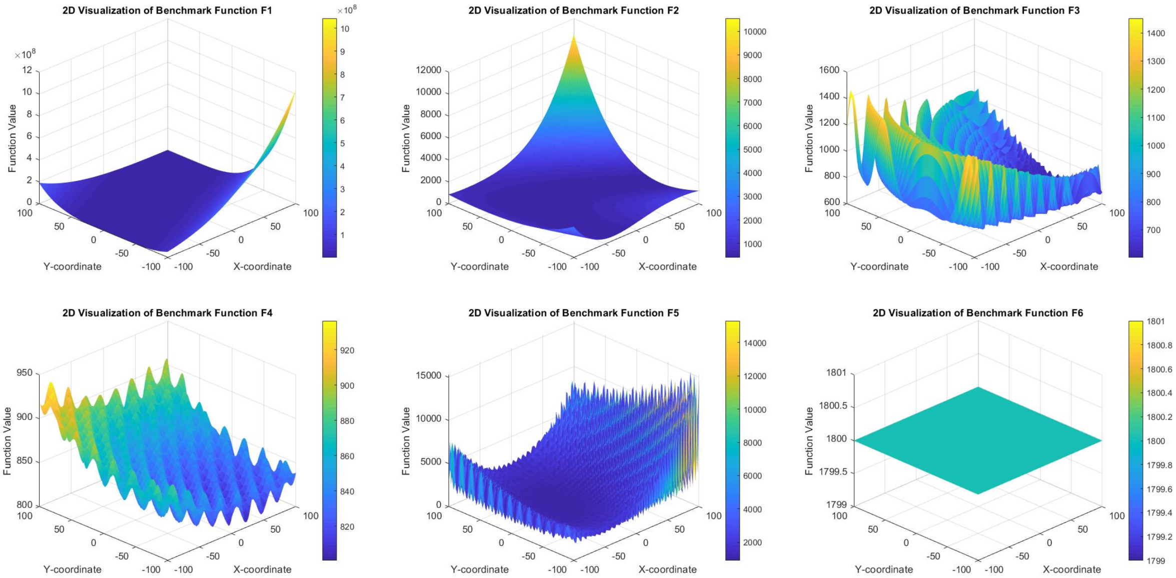

Figure 2. Illustration of CEC2022 benchmark functions (F1–F6).

Structured to ensure fairness, impartiality, and a reflection of real-world complexities, the CEC2022 benchmarks are suitable for testing the COASaDE optimizer. This suite allows researchers to test their algorithms against cutting-edge methods in a competitive setting that mirrors current industry and academic demands. The CEC 2022 Special Session and Competition on Single-Objective Bound-Constrained Numerical Optimization offers a pivotal platform for the COASaDE optimizer to demonstrate its efficacy and for researchers to identify potential improvements and inspire the development of next-generation optimization technologies.

5.2. CEC2017 Benchmark Description

The CEC2017 benchmark suite serves as an exhaustive testing ground for the proposed COASaDE optimizer, comprising a diverse array of functions that present multifaceted challenges typical of real-world optimization scenarios. This suite includes multimodal landscapes, high-dimensional spaces, and intricate global structures that rigorously test an algorithm's capability to efficiently explore and exploit the search space. These functions are strategically chosen to mimic real-life problems, ensuring that the insights gleaned from the COASaDE's performance are relevant and applicable in practical applications.

Each function within the CEC2017 suite is designed to highlight specific challenges, such as navigating extensive flat areas without premature convergence and handling noise that can obscure the true optima. Additionally, the inclusion of rotated and shifted versions of the base functions introduces further complexity, simulating the effects of variable data transformations seen in real-world environments. These features make the CEC2017 benchmarks suitable for assessing the COASaDE optimizer′s precision, scalability, and adaptability across different orientations and problem scales, thereby offering a thorough evaluation of its practical applicability and robustness.

5.3. Configuration of Parameters for CEC Benchmark Evaluations

The parameters utilized for comparing the optimizers are detailed in Table 1. A consistent parameter configuration is crucial for ensuring fair and effective evaluation across the CEC benchmarks from 2022, 2017, and 2014 and for facilitating a standardized testing environment across various functions.

Table 1. Configuration of parameters across CEC benchmarks.

Parameter Type	Assigned Value
Size of Population	30 individuals
Evaluation Limit	1000 iterations
Problem Dimensions (D)	10
Boundary of Search	$[-100, 100]^D$
Incorporation of Rotation	Applied where specified
Application of Shift	Applied where specified

The parameters listed in Table 1 were adopted to create a balanced framework supporting comparisons across diverse studies, ensuring a comprehensive and rigorous performance assessment of the algorithms. The chosen population size of 100 and problem dimensionality of 30 strike an optimal balance between the computational demand and the complexity required for significant evaluations. The ceiling on function evaluations, defined as 10,000 × D, provides ample opportunities for the algorithms to demonstrate their efficiency and convergence capabilities over extensive iterations. The fixed search boundary, set from $[-100, 100]$ across all dimensions, ensures a vast and uniform exploration space. Finally, specific benchmarks were made more challenging with the application of rotational and shifting transformations to test the adaptability and robustness of algorithms under enhanced complexity conditions.

A comprehensive statistical analysis was employed for the performance evaluation of various optimization algorithms (see Table 2), including calculation of the mean and standard deviation and the application of the Wilcoxon rank-sum test. The mean serves as a measure of the central tendency, representing the average result obtained by the algorithms across multiple trials. In contrast, the standard deviation quantifies the dispersion or variability around this mean, offering insights into the consistency and reliability of the algorithms across different runs.

Table 2. Parameter settings for the compared algorithms.

Algorithm	Parameter
MFO	a linearly decreases from −1 to −2
GWO	Convergence constant a = [2, 0]
PSO	$V_{\max} = 6$, $W_{\max} = 0.9$, $W_{\min} = 0.2$, $C_1 = C_2 = 2$
MVO	Modular modality = 0.01, Power exponent = v, Switch probability = 0.8
AOA	$\alpha = 5$, Mu = 0.499
MVO	$WEP_{\text{Max}} = 1$, $WEP_{\text{Min}} = 0.2$
SHO	U = 0.05, V = 0.05, L = 0.05
SCA	$r1 = \text{random}(0, 1)$, $r2 = \text{random}(0, 1)$, $r3 = \text{random}(0, 1)$, $r4 = \text{random}(0, 1)$

To further assess the statistical significance of differences between the performance of two distinct groups of algorithmic outcomes, the Wilcoxon rank-sum test was utilized. This nonparametric test helps to determine whether there is a statistically significant difference between the medians of the two groups, providing a robust method for comparing the performance of optimization algorithms under various conditions.

5.4. CEC2022 Results

The performance of COASaDE on the CEC2022 benchmark suite demonstrates its robustness and superiority compared to several well known and state-of-the-art optimization algorithms. As shown in Table 3, COASaDE consistently ranks highly across various test functions, often achieving the best or near-best mean and standard deviation values. For instance, on Function F1, COASaDE achieves the lowest mean (3.00E+02) and smallest standard deviation (1.42E-04), significantly outperforming other algorithms including GWO, PSO, and MFO, all of which have higher means and larger standard deviations. This trend continues across multiple functions, indicating COASaDE's ability to maintain stable and high quality solutions. The algorithm's self-adaptive mechanism for adjusting mutation factors and crossover rates plays a crucial role in its success, enabling it to effectively balance exploration and exploitation. The results suggest that COASaDE's combination of crayfish-inspired foraging behavior and differential evolution techniques makes it a powerful tool for tackling complex optimization problems.

Table 3. Comparison results with well known and state-of-the-art optimizers on CEC2022, FES = 1000, Agents = 50.

Function	Statistics	COASaDE	GWO	PSO	MFO	MVO	SHIO	OHO	AOA	FOX	FVIM	SCA
F1	Mean	**3.00E+02**	8.90E+02	1.58E+04	1.09E+04	8.80E+03	6.68E+03	1.20E+04	6.32E+03	1.17E+03	7.44E+02	1.39E+03
	Std	1.42E-04	1.05E+03	5.10E+03	9.50E+03	4.51E+03	3.42E+03	2.05E+03	2.54E+03	4.53E+02	6.02E+02	1.41E+03
	SEM	6.34E-05	4.70E+02	2.28E+03	4.25E+03	2.02E+03	1.53E+03	9.18E+02	1.14E+03	2.03E+02	2.69E+02	6.31E+02
	Rank	1	3	11	9	8	7	10	6	4	2	5
F2	Mean	**4.05E+02**	4.26E+02	4.14E+02	4.10E+02	1.71E+03	4.49E+02	3.03E+03	8.06E+02	4.55E+02	4.42E+02	4.61E+02
	Std	1.94E+00	3.07E+01	1.38E+01	5.29E+00	5.52E+02	3.73E+01	1.41E+03	1.01E+02	1.24E+01	2.28E+01	3.46E+01
	SEM	8.68E-01	1.37E+01	6.19E+00	2.36E+00	2.47E+02	1.67E+01	6.29E+02	4.53E+01	5.54E+00	1.02E+01	1.55E+01
	Rank	1	4	3	2	10	6	11	9	7	5	8
F3	Mean	**6.00E+02**	6.00E+02	6.33E+02	6.02E+02	6.42E+02	6.05E+02	6.55E+02	6.36E+02	6.14E+02	6.05E+02	6.15E+02
	Std	0.00E+00	3.33E-01	1.13E+01	2.55E+00	5.20E+00	5.29E+00	6.88E+00	6.04E+00	2.75E+00	3.24E+00	8.51E+00
	SEM	0.00E+00	1.49E-01	5.06E+00	1.14E+00	2.33E+00	2.36E+00	3.08E+00	2.70E+00	1.23E+00	1.45E+00	3.80E+00
	Rank	1	2	8	3	10	5	11	9	6	4	7
F4	Mean	8.27E+02	8.15E+02	8.35E+02	8.35E+02	8.50E+02	**8.12E+02**	8.48E+02	8.30E+02	8.37E+02	8.28E+02	8.28E+02
	Std	3.29E+00	5.10E+00	1.26E+01	3.27E+00	5.63E+00	2.85E+00	2.96E+00	3.61E+00	7.36E+00	6.65E+00	8.70E+00
	SEM	1.47E+00	2.28E+00	5.65E+00	1.46E+00	2.52E+00	1.28E+00	1.32E+00	1.62E+00	3.29E+00	2.97E+00	3.89E+00
	Rank	3	2	7	8	11	1	10	6	9	5	4
F5	Mean	**9.00E+02**	9.01E+02	1.32E+03	9.77E+02	1.33E+03	9.92E+02	1.55E+03	1.31E+03	9.99E+02	9.52E+02	1.05E+03
	Std	0.00E+00	8.21E-01	2.83E+02	1.31E+02	8.38E+01	1.01E+02	4.61E+01	1.32E+02	1.83E+01	3.92E+01	1.26E+02
	SEM	0.00E+00	3.67E-01	1.27E+02	5.85E+01	3.75E+01	4.51E+01	2.06E+01	5.90E+01	8.18E+00	1.76E+01	5.61E+01
	Rank	1	2	9	4	10	5	11	8	6	3	7
F6	Mean	**1.80E+03**	5.71E+03	2.74E+03	5.72E+03	4.59E+07	2.99E+03	1.26E+09	3.68E+03	1.65E+06	1.01E+04	4.56E+03
	Std	5.26E-01	3.35E+03	1.38E+03	2.53E+03	7.73E+07	1.06E+03	9.46E+08	8.62E+02	1.18E+06	2.55E+03	1.76E+03
	SEM	2.35E-01	1.50E+03	6.18E+02	1.13E+03	3.46E+07	4.72E+02	4.23E+08	3.86E+02	5.27E+05	1.14E+03	7.89E+02
	Rank	1	6	2	7	10	3	11	4	9	8	5
F7	Mean	**2.00E+03**	2.03E+03	2.06E+03	2.03E+03	2.09E+03	2.05E+03	2.12E+03	2.11E+03	2.05E+03	2.05E+03	2.03E+03
	Std	9.39E+00	2.35E+01	2.10E+01	1.65E+01	1.75E+01	1.88E+01	1.06E+01	1.99E+01	5.12E+00	1.79E+01	1.59E+01
	SEM	4.20E+00	1.05E+01	9.40E+00	7.37E+00	7.81E+00	8.40E+00	4.76E+00	8.88E+00	2.29E+00	8.00E+00	7.12E+00
	Rank	1	2	8	3	9	6	11	10	7	5	4
F8	Mean	**2.20E+03**	2.22E+03	2.24E+03	2.22E+03	2.33E+03	2.23E+03	2.37E+03	2.29E+03	2.23E+03	2.23E+03	2.22E+03
	Std	1.05E+00	2.24E+00	1.80E+01	1.96E+00	8.44E+01	5.01E+00	6.65E+01	1.03E+02	2.94E+00	6.57E-01	2.28E+00
	SEM	4.71E-01	1.00E+00	8.07E+00	8.75E-01	3.77E+01	2.24E+00	2.97E+01	4.61E+01	1.31E+00	2.94E-01	1.02E+00
	Rank	1	2	8	3	10	6	11	9	7	5	4
F9	Mean	**2.53E+03**	2.54E+03	2.55E+03	2.53E+03	2.79E+03	2.56E+03	2.82E+03	2.70E+03	2.57E+03	2.58E+03	2.57E+03
	Std	0.00E+00	1.86E+01	2.88E+01	2.49E-01	8.91E+01	3.38E+01	3.82E+01	2.23E+01	2.04E+01	3.20E+01	2.03E+01
	SEM	0.00E+00	8.31E+00	1.29E+01	1.11E-01	3.98E+01	1.51E+01	1.71E+01	9.96E+00	9.13E+00	1.43E+01	9.06E+00
	Rank	1	3	4	2	10	5	11	9	6	8	7

Table 3. *Cont.*

Function	Statistics	COASaDE	GWO	PSO	MFO	MVO	SHIO	OHO	AOA	FOX	FVIM	SCA
F10	Mean	**2.50E+03**	2.57E+03	2.66E+03	2.50E+03	2.50E+03	2.53E+03	3.14E+03	2.74E+03	2.50E+03	2.55E+03	2.58E+03
	Std	5.96E-02	6.05E+01	1.99E+02	1.24E+00	1.08E+00	5.54E+01	4.27E+02	2.00E+02	4.05E-01	6.42E+01	6.89E+01
	SEM	2.67E-02	2.71E+01	8.89E+01	5.55E-01	4.81E-01	2.48E+01	1.91E+02	8.95E+01	1.81E-01	2.87E+01	3.08E+01
	Rank	1	7	9	2	4	5	11	10	3	6	8
F11	Mean	**2.60E+03**	2.87E+03	2.84E+03	2.70E+03	2.99E+03	2.83E+03	3.69E+03	3.20E+03	2.77E+03	2.94E+03	2.72E+03
	Std	3.94E-13	2.12E+02	1.20E+02	9.31E+01	1.30E+02	2.64E+02	7.01E+02	3.10E+02	4.19E+00	2.37E+02	5.71E+01
	SEM	1.76E-13	9.46E+01	5.37E+01	4.16E+01	5.81E+01	1.18E+02	3.14E+02	1.38E+02	1.87E+00	1.06E+02	2.56E+01
	Rank	1	7	6	2	9	5	11	10	4	8	3
F12	Mean	**2.86E+03**	2.87E+03	2.89E+03	2.86E+03	2.90E+03	2.88E+03	3.09E+03	2.99E+03	2.87E+03	2.87E+03	2.89E+03
	Std	7.50E-01	2.48E+00	3.03E+01	7.52E-01	2.25E+01	1.43E+01	4.32E+01	3.01E+01	1.73E+00	3.40E+00	2.20E+01
	SEM	3.35E-01	1.11E+00	1.36E+01	3.36E-01	1.01E+01	6.37E+00	1.93E+01	1.34E+01	7.72E-01	1.52E+00	9.85E+00
	Rank	1	3	8	2	9	6	11	10	5	4	7

The performance of COASaDE on the CEC2022 benchmark suite is shown in Table 4, which shows the comparison results with various Differential Evolution (DE) variants. The results highlight its effectiveness and competitiveness, showing that COASaDE consistently ranks among the top performers across multiple test functions. For Function F1, COASaDE achieves the best mean (3.00E+02) and smallest standard deviation (1.42E-04), demonstrating its precision and stability compared to other DE variants such as LDE, BBDE, and JADE, which all have higher means and larger standard deviations. This trend of superior performance continues with Functions F2 and F3, where COASaDE secures the top ranks, indicating its robustness in handling different optimization challenges. COASaDE also performs well on Function F4, showcasing its adaptability, although it ranks slightly behind some variants.

Table 4. Comparison Results with variant Differential Evolution optimizers on CEC2022, FES = 1000, Agents = 50.

Function	Statistics	COASaDE	COA	LDE	BBDE	ODE	JADE	DEEM	SADE	CMAES
F1	Mean	**3.00E+02**	3.01E+02	3.18E+02	8.25E+02	3.00E+02	3.64E+02	3.01E+02	7.65E+02	2.28E+04
	Std	1.42E-04	6.39E-01	1.09E+01	6.10E+02	1.38E-03	8.27E+01	0.00E+00	2.53E+02	1.08E+04
	SEM	6.34E-05	2.86E-01	4.88E+00	2.73E+02	6.17E-04	3.70E+01	0.00E+00	1.13E+02	4.81E+03
	Rank	1	4	5	8	2	6	3	7	9
F2	Mean	**4.05E+02**	4.06E+02	4.08E+02	4.07E+02	4.08E+02	4.07E+02	4.06E+02	4.08E+02	5.80E+02
	Std	1.94E+00	3.50E+00	1.76E+00	1.64E+00	1.87E+00	2.64E+00	4.05E+00	7.99E-01	6.89E+01
	SEM	8.68E-01	1.57E+00	7.87E-01	7.34E-01	8.34E-01	1.18E+00	1.81E+00	3.57E-01	3.08E+01
	Rank	1	2	6	5	7	4	3	8	9
F3	Mean	**6.00E+02**	6.07E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.28E+02
	Std	0.00E+00	1.35E+01	3.16E-02	8.04E-14	0.00E+00	7.85E-07	5.32E-02	0.00E+00	2.98E+01
	SEM	0.00E+00	6.03E+00	1.41E-02	3.60E-14	0.00E+00	3.51E-07	2.38E-02	0.00E+00	1.33E+01
	Rank	1	8	7	1	1	5	6	1	9
F4	Mean	8.27E+02	8.33E+02	8.30E+02	8.12E+02	8.30E+02	**8.06E+02**	8.11E+02	8.28E+02	8.27E+02
	Std	3.29E+00	4.45E-01	3.85E+00	5.23E+00	1.74E+00	8.22E-01	7.22E+00	5.32E+00	1.04E+01
	SEM	1.47E+00	1.99E-01	1.72E+00	2.34E+00	7.80E-01	3.68E-01	3.23E+00	2.38E+00	4.65E+00
	Rank	4	9	8	3	7	1	2	6	5
F5	Mean	**900**	902.4721	900.2119	900.0179	900	900.0001	900.1088	900.0001	900
	Std	0	4.641643	0.046755	0.040038	0	7.37E-05	0.243218	0.000172	0
	SEM	0	2.075806	0.02091	0.017906	0	3.3E-05	0.10877	7.68E-05	0
	Rank	1	9	8	6	1	4	7	5	1
F6	Mean	**1.80E+03**	3.64E+03	5.98E+03	2.77E+03	1.80E+03	1.85E+03	1.81E+03	1.84E+03	7.61E+07
	Std	5.26E-01	2.56E+03	2.80E+03	1.08E+03	4.98E-01	2.79E+01	5.82E+00	1.47E+01	1.10E+08
	SEM	2.35E-01	1.14E+03	1.25E+03	4.83E+02	2.23E-01	1.25E+01	2.60E+00	6.57E+00	4.92E+07
	Rank	1	7	8	6	2	5	3	4	9
F7	Mean	2.00E+03	2.02E+03	2.02E+03	2.02E+03	**2.00E+03**	2.01E+03	2.02E+03	2.00E+03	2.09E+03
	Std	9.39E+00	1.11E+00	1.71E+00	6.83E-01	2.70E-01	1.06E+01	8.45E+00	1.52E+00	5.05E+01
	SEM	4.20E+00	4.96E-01	7.66E-01	3.06E-01	1.21E-01	4.76E+00	3.78E+00	6.80E-01	2.26E+01
	Rank	2	6	8	7	1	4	5	3	9
F8	Mean	**2.20E+03**	2.23E+03	2.22E+03	2.22E+03	2.20E+03	2.21E+03	2.21E+03	2.21E+03	2.25E+03
	Std	1.05E+00	2.61E+00	5.99E+00	2.28E+00	9.85E-01	8.33E+00	1.12E+01	2.95E+00	6.40E+00
	SEM	4.71E-01	1.17E+00	2.68E+00	1.02E+00	4.41E-01	3.72E+00	5.01E+00	1.32E+00	2.86E+00
	Rank	1	8	6	7	2	5	4	3	9

Table 4. *Cont.*

Function	Statistics	COASaDE	COA	LDE	BBDE	ODE	JADE	DEEM	SADE	CMAES
F9	Mean	**2.53E+03**	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.57E+03
	Std	0.00E+00	6.15E-06	1.82E-03	0.00E+00	0.00E+00	4.57E-05	0.00E+00	0.00E+00	4.22E+01
	SEM	0	2.75E-06	0.000815	0	0	2.04E-05	0	0	18.86533
	Rank	1	6	8	1	1	7	1	1	9
F10	Mean	2.50E+03	2.53E+03	2.50E+03	2.50E+03	**2.50E+03**	2.52E+03	2.50E+03	2.50E+03	2.68E+03
	Std	5.96E-02	5.74E+01	5.12E-02	8.11E-02	1.03E-02	4.68E+01	1.45E-01	6.80E-02	2.22E+02
	SEM	2.67E-02	2.57E+01	2.29E-02	3.63E-02	4.61E-03	2.09E+01	6.50E-02	3.04E-02	9.93E+01
	Rank	2	8	4	6	1	7	5	3	9
F11	Mean	**2.60E+03**	2.75E+03	2.61E+03	2.63E+03	2.60E+03	2.68E+03	2.80E+03	2.63E+03	3.13E+03
	Std	3.94E-13	1.50E+02	7.91E-01	6.73E+01	3.22E-13	1.79E+02	1.54E+02	6.73E+01	2.58E+02
	SEM	1.76E-13	6.71E+01	3.54E-01	3.01E+01	1.44E-13	7.99E+01	6.89E+01	3.01E+01	1.15E+02
	Rank	1	7	3	4	1	6	8	5	9
F12	Mean	**2.86E+03**	2.87E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03	2.87E+03
	Std	7.50E-01	1.18E+00	1.04E+00	6.62E-01	2.04E+00	6.71E-01	2.03E+00	9.78E-01	2.37E+00
	SEM	3.35E-01	5.27E-01	4.67E-01	2.96E-01	9.11E-01	3.00E-01	9.09E-01	4.38E-01	1.06E+00
	Rank	1	8	2	5	4	7	3	6	9

Moreover, the results indicate that COASaDE maintains a strong balance between exploration and exploitation, facilitated by its adaptive mechanisms for mutation and crossover rates. For Functions F6 to F12 COASaDE consistently ranks within the top positions, often securing the first or second rank. The algorithm's ability to dynamically adjust its parameters based on the current state of the optimization process allows it to effectively navigate the search space and avoid local optima. Additionally, the comparative analysis with DE variants such as SADE, CMAES, and others highlights COASaDE's competitive edge in terms of mean performance and consistency, as reflected in the values for the standard error of the mean (SEM). Overall, the results from Table 2 underscore COASaDE's superior optimization capabilities, robust performance, and effectiveness as a hybrid optimizer that successfully integrates heuristic strategies with mathematical rigor.

5.5. Wilcoxon Rank-Sum Test Results

The Wilcoxon rank-sum test results highlight the statistical significance of COASaDE's performance on the CEC2022 benchmark functions compared to various other optimizers. Across multiple functions in Tables 5 and 6 (F1 to F12), COASaDE consistently achieves p-values of 3.97×10^{-3} against most compared optimizers, indicating statistically significant improvement. For example, on Function F1 COASaDE shows significant differences with all listed algorithms, including GWO, PSO, and MFO. On Function F2, while maintaining strong significance against most optimizers, it exhibits less significant results compared to SHIO and slightly higher p-values with AOA and SCA. This pattern is observed across other functions as well, with COASaDE demonstrating robust performance and often outperforming competitors by statistically significant margins. The few instances of higher p-values, such as in F6 with SHIO and SCA and in F10 with GWO and SHIO, suggest areas where the performance differences are less pronounced. Overall, the Wilcoxon test results affirm the effectiveness and reliability of COASaDE in providing superior optimization results across various challenging functions.

Table 5. Wilcoxon rank-sum test over CEC2022, FES = 1000, Agents = 50.

Fun	GWO	PSO	MFO	MVO	SHIO	OHO	AOA	FOX	SCA
F1	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03
F2	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	9.96E-01	1.59E-02	3.97E-03	7.94E-03
F3	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03
F4	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03
F5	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03
F6	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	9.84E-01	3.97E-03	3.97E-03	7.26E-01
F7	3.97E-03	7.94E-03	1.59E-02	3.97E-03	3.97E-03	3.45E-01	1.55E-01	3.97E-03	2.74E-01
F8	3.97E-03	2.78E-02	3.97E-03	3.97E-03	3.97E-03	1.00E+00	7.54E-02	3.97E-03	7.54E-02
F9	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	7.26E-01	3.97E-03	3.97E-03	2.10E-01
F10	2.74E-01	5.00E-01	1.11E-01	3.97E-03	2.78E-02	9.84E-01	7.90E-01	1.11E-01	6.55E-01
F11	3.97E-03	3.97E-03	3.97E-03	3.97E-03	2.78E-02	3.45E-01	2.74E-01	7.94E-03	2.78E-02
F12	1.59E-02	7.90E-01	3.97E-03	3.97E-03	9.92E-01	1.00E+00	1.00E+00	2.10E-01	1.00E+00

Table 6. Wilcoxon rank-sum test over CEC2022, FES = 1000, Agents = 50.

Fun	LDE	BBDE	ODE	CCDE	JADE	DEEM	SADE	JADE	CMAES
F1	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03
F2	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03
F3	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	1.59E-02
F4	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03
F5	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03
F6	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.45E-01
F7	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	7.54E-02
F8	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	7.94E-03
F9	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03
F10	3.97E-03	3.97E-03	3.97E-03	3.97E-03	2.78E-02	3.97E-03	3.97E-03	3.97E-03	4.21E-01
F11	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	1.11E-01
F12	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	3.97E-03	9.96E-01

5.6. COASaDE Results on CEC2017

The results of COASaDE on the CEC2017 benchmark suite, shown in Table 7, highlight its strong performance relative to other optimizers. For Function F1, COASaDE achieves the lowest mean value of 6.85×10^9 and ranks first, demonstrating its efficiency in solving this optimization problem compared to other algorithms such as GWO, PSO, and MFO, which have significantly higher mean values. Similarly, on Function F2 COASaDE ranks third with a mean value of 2.30×10^{12}, showing competitive performance among the top optimizers. On Function F3, COASaDE excels with a mean value of 4.12×10^4, securing the third position and outperforming algorithms such as MFO and GWO, both of which rank lower. For Function F4, COASaDE again ranks first, with a mean value of 1.09×10^3, indicating its robustness and stability. The algorithm continues to perform well on Function F5, ranking first with a mean value of 6.13×10^2, highlighting its consistent ability to find optimal solutions. The trend of strong performance is evident across other functions as well. For example, on Function F6 COASaDE achieves the top rank with a mean value of 6.59×10^2. On Function F7, it ranks third with a mean value of 1.01×10^3, showcasing its competitive edge. On Function F8, COASaDE ranks second with a mean value of 8.81×10^2, demonstrating its precision and effectiveness.

On more challenging functions such as F9 and F10, COASaDE maintains its competitive performance, ranking third and second, respectively. Its mean values are 3.69×10^3 for F9 and 3.28×10^3 for F10. This consistency is further evident on Function F11, where COASaDE ranks third with a mean value of 4.30×10^3, outperforming many other well-known algorithms. For Function F12, COASaDE ranks second with a mean value of 6.95×10^8, reinforcing its strong optimization capabilities. On Function F13 it ranks third with a mean value of 3.24×10^7, showcasing its adaptability. On Function F14, COASaDE secures the second position with a mean value of 1.94×10^4, demonstrating its robustness across diverse optimization problems. Finally, on Function F15 COASaDE ranks fifth with a mean value of 2.78×10^5, indicating that while it is highly competitive there is still room for improvement in certain complex scenarios.

Table 7. Comparison test results with different optimizers on CEC2017, FES = 1000, Agents = 50.

Fun	Statistics	COASaDE	GWO	PSO	MFO	MVO	SHIO	OHO	FOX	FVIM	SCA
F1	Mean	**6.85E+09**	1.65E+10	8.00E+09	1.35E+10	1.67E+10	1.29E+10	1.09E+10	1.63E+10	1.61E+10	1.47E+10
	Std	1.65E+09	3.62E+09	2.29E+09	4.24E+09	7.28E+09	3.34E+09	3.21E+09	7.03E+09	3.37E+09	4.35E+09
	SEM	7.36E+08	1.62E+09	1.02E+09	1.90E+09	3.25E+09	1.50E+09	1.44E+09	3.14E+09	1.51E+09	1.95E+09
	Rank	1	9	2	5	10	4	3	8	7	6
F2	Mean	2.30E+12	3.13E+12	**7.20E+11**	6.69E+12	8.99E+12	9.80E+12	1.32E+13	1.27E+14	2.96E+13	1.18E+12
	Std	3.80E+12	4.66E+12	1.13E+12	6.28E+12	1.33E+13	1.68E+13	1.84E+13	2.78E+14	4.21E+13	1.59E+12
	SEM	1.70E+12	2.09E+12	5.04E+11	2.81E+12	5.95E+12	7.52E+12	8.23E+12	1.24E+14	1.88E+13	7.13E+11
	Rank	3	4	1	5	6	7	8	10	9	2

Table 7. *Cont.*

Fun	Statistics	COASaDE	GWO	PSO	MFO	MVO	SHIO	OHO	FOX	FVIM	SCA
F3	Mean	4.12E+04	4.39E+04	7.84E+04	5.24E+04	7.87E+04	5.22E+04	**2.26E+04**	2.64E+04	4.24E+04	6.16E+04
	Std	7.94E+03	1.23E+04	3.57E+04	2.09E+04	3.92E+04	2.24E+04	6.82E+03	1.55E+04	2.69E+04	9.12E+03
	SEM	3.55E+03	5.49E+03	1.60E+04	9.35E+03	1.75E+04	1.00E+04	3.05E+03	6.95E+03	1.20E+04	4.08E+03
	Rank	3	5	9	7	10	6	1	2	4	8
F4	Mean	**1.09E+03**	1.57E+03	1.25E+03	1.74E+03	1.28E+03	1.88E+03	1.66E+03	1.85E+03	1.41E+03	1.56E+03
	Std	2.01E+02	3.31E+02	5.51E+02	2.98E+02	4.19E+02	6.57E+02	2.82E+02	8.11E+02	4.69E+02	4.27E+02
	SEM	8.98E+01	1.48E+02	2.46E+02	1.33E+02	1.88E+02	2.94E+02	1.26E+02	3.63E+02	2.10E+02	1.91E+02
	Rank	1	6	2	8	3	10	7	9	4	5
F5	Mean	**6.13E+02**	6.31E+02	6.20E+02	6.43E+02	6.29E+02	6.17E+02	6.14E+02	6.24E+02	6.21E+02	6.26E+02
	Std	8.74E+00	1.80E+01	2.37E+01	7.98E+00	2.39E+01	1.66E+01	1.79E+01	1.10E+01	1.48E+01	3.58E+00
	SEM	3.91E+00	8.06E+00	1.06E+01	3.57E+00	1.07E+01	7.42E+00	8.00E+00	4.90E+00	6.62E+00	1.60E+00
	Rank	1	9	4	10	8	3	2	6	5	7
F6	Mean	**6.59E+02**	6.69E+02	6.61E+02	6.69E+02	6.74E+02	6.75E+02	6.63E+02	6.68E+02	6.62E+02	6.83E+02
	Std	1.33E+01	1.02E+01	1.93E+01	1.09E+01	7.17E+00	1.88E+01	8.29E+00	5.39E+00	6.06E+00	9.77E+00
	SEM	5.93E+00	4.56E+00	8.63E+00	4.86E+00	3.21E+00	8.42E+00	3.71E+00	2.41E+00	2.71E+00	4.37E+00
	Rank	1	7	2	6	8	9	4	5	3	10
F7	Mean	1.01E+03	1.01E+03	1.02E+03	1.11E+03	8.52E+02	1.01E+03	**8.20E+02**	1.01E+03	1.01E+03	1.07E+03
	Std	2.59E+01	6.54E+01	2.49E+01	8.23E+01	4.50E+00	2.86E+01	3.03E+01	3.49E+01	2.71E+01	9.70E+01
	SEM	1.16E+01	2.93E+01	1.11E+01	3.68E+01	2.01E+00	1.28E+01	1.36E+01	1.56E+01	1.21E+01	4.34E+01
	Rank	3	4	8	10	2	7	1	6	5	9
F8	Mean	8.81E+02	9.03E+02	8.91E+02	9.12E+02	8.95E+02	8.83E+02	**8.68E+02**	8.87E+02	8.90E+02	8.99E+02
	Std	7.28E+00	7.40E+00	1.36E+01	1.68E+01	1.15E+01	1.40E+01	8.01E+00	7.17E+00	2.25E+01	1.67E+01
	SEM	3.26E+00	3.31E+00	6.08E+00	7.51E+00	5.14E+00	6.25E+00	3.58E+00	3.21E+00	1.01E+01	7.45E+00
	Rank	2	9	6	10	7	3	1	4	5	8
F9	Mean	3.69E+03	3.75E+03	3.71E+03	4.57E+03	2.06E+03	3.76E+03	**1.84E+03**	3.71E+03	3.73E+03	3.83E+03
	Std	1.28E+03	5.35E+02	1.00E+03	6.99E+02	2.76E+02	3.86E+02	4.24E+02	8.82E+02	1.17E+03	6.68E+02
	SEM	5.72E+02	2.39E+02	4.49E+02	3.13E+02	1.24E+02	1.72E+02	1.90E+02	3.94E+02	5.25E+02	2.99E+02
	Rank	3	7	4	10	2	8	1	5	6	9
F10	Mean	3.28E+03	3.34E+03	**3.05E+03**	3.50E+03	3.32E+03	3.31E+03	3.30E+03	3.34E+03	3.36E+03	3.64E+03
	Std	5.27E+01	2.09E+02	1.62E+02	3.16E+02	2.40E+02	2.07E+02	3.03E+02	2.17E+02	3.41E+02	2.05E+02
	SEM	2.36E+01	9.36E+01	7.25E+01	1.41E+02	1.07E+02	9.25E+01	1.36E+02	9.71E+01	1.52E+02	9.18E+01
	Rank	2	6	1	9	5	4	3	7	8	10
F11	Mean	4.30E+03	**3.35E+03**	1.34E+04	1.28E+04	9.75E+03	4.31E+03	4.63E+03	4.12E+03	5.85E+03	5.51E+03
	Std	1.94E+03	1.45E+03	8.35E+03	6.83E+03	6.19E+03	1.97E+03	2.08E+03	2.39E+03	1.48E+03	2.71E+03
	SEM	8.68E+02	6.49E+02	3.73E+03	3.05E+03	2.77E+03	8.79E+02	9.32E+02	1.07E+03	6.62E+02	1.21E+03
	Rank	3	1	10	9	8	4	5	2	7	6
F12	Mean	6.95E+08	1.37E+09	8.01E+08	9.42E+08	1.27E+09	9.35E+08	**5.73E+08**	1.36E+09	1.31E+09	1.05E+09
	Std	3.38E+08	1.11E+09	5.16E+08	4.54E+08	7.85E+07	5.00E+08	2.58E+08	5.87E+08	5.28E+08	4.55E+08
	SEM	1.51E+08	4.98E+08	2.31E+08	2.03E+08	3.51E+07	2.24E+08	1.15E+08	2.62E+08	2.36E+08	2.03E+08
	Rank	2	10	3	5	7	4	1	9	8	6
F13	Mean	3.24E+07	3.31E+07	**6.79E+05**	3.48E+07	8.47E+07	3.45E+07	3.85E+07	9.84E+07	1.05E+08	4.50E+06
	Std	2.02E+07	2.62E+07	7.25E+05	2.68E+07	8.23E+07	2.16E+07	1.22E+07	7.32E+07	1.18E+08	4.28E+06
	SEM	9.06E+06	1.17E+07	3.24E+05	1.20E+07	3.68E+07	9.67E+06	5.43E+06	3.27E+07	5.26E+07	1.92E+06
	Rank	3	4	1	6	8	5	7	9	10	2
F14	Mean	1.94E+04	6.55E+04	**1.10E+04**	5.31E+05	5.31E+05	9.32E+05	1.44E+05	1.65E+05	5.90E+05	1.92E+05
	Std	2.27E+04	8.24E+04	1.07E+04	1.07E+06	4.78E+05	6.52E+05	1.51E+05	1.51E+05	6.84E+05	1.26E+05
	SEM	1.01E+04	3.68E+04	4.79E+03	4.80E+05	2.14E+05	2.91E+05	6.74E+04	6.76E+04	3.06E+05	5.62E+04
	Rank	2	3	1	7	8	10	4	5	9	6
F15	Mean	2.78E+05	6.15E+04	6.65E+04	6.20E+05	8.90E+06	2.90E+05	6.66E+04	**5.95E+04**	3.18E+06	3.05E+05
	Std	3.34E+05	9.78E+04	6.39E+04	1.21E+06	1.45E+07	2.87E+05	7.60E+04	3.94E+04	4.77E+06	5.06E+05
	SEM	1.49E+05	4.37E+04	2.86E+04	5.40E+05	6.50E+06	1.28E+05	3.40E+04	1.76E+04	2.13E+06	2.26E+05
	Rank	5	2	3	8	10	6	4	1	9	7

The performance of COASaDE on CEC2017 functions F16 to F30, shown in Table 8, indicates its strong competitiveness across a variety of optimization problems. For F16, COASaDE ranks first with a mean value of 2.25 × 10^3, outperforming all other algorithms, including GWO and PSO. On F17, COASaDE ranks third with a mean value of 2.09 × 10^3, demonstrating its consistent performance among the top optimizers. Notably, COASaDE achieves the lowest mean value on F18 with 1.21 × 10^7, highlighting its exceptional ability to handle complex optimization scenarios. On F19, COASaDE again secures the top position with a mean of 5.85 × 10^5, showcasing its robustness in solving high-dimensional functions. It maintains this trend on F20, ranking second with a mean value of 2.31 × 10^3. For F21, COASaDE ranks second with a mean value of 2.39 × 10^3, further indicating its reliability. Similarly, COASaDE's performance in F22 is commendable; it achieves the lowest mean value of 3.08 × 10^3, outperforming other algorithms by a significant margin. The trend continues for F23, where COASaDE ranks first with a mean value of 2.71 × 10^3, and for F24, where it secures the top rank with a mean of 2.85 × 10^3. On F25, COASaDE ranks second with a mean value of 3.58 × 10^3, demonstrating its ability to consistently find optimal solutions across different problems. For F26, COASaDE ranks second with a mean value of 4.56 × 10^3, confirming its effectiveness in various optimization contexts. On F27, COASaDE ranks first with a mean value of 3.19 × 10^3, outperforming other well-known algorithms such as GWO and PSO. For F28, it achieves the lowest mean value of 3.62 × 10^3, highlighting its superior performance. COASaDE also excels on F29, securing the top position with a mean value of 3.67 × 10^3. Lastly, COASaDE ranks first on F30 with a mean value of 2.56 × 10^7, outperforming all other algorithms and demonstrating its capability to effectively handle complex and diverse optimization problems.

As can be seen from Table 9, the comparison of COASaDE with other Differential Evolution optimizers on CEC2017 functions F1 to F15 demonstrates strong performance across a variety of optimization problems. For F1, COASaDE ranks second with a mean value of 6.85 × 10^9, closely following JADE, which ranks first. On F2, COASaDE ranks third with a mean of 2.30 × 10^{12}, outperformed only by JADE and DEEM. COASaDE achieves the best performance on F3, ranking first with a mean of 4.12 × 10^4, demonstrating its efficiency in solving this problem. On F4, COASaDE again secures the top position with a mean value of 1.05 × 10^3, indicating its robustness. For F5, COASaDE ranks fifth with a mean of 6.13 × 10^2, showing competitive, though not the best performance. COASaDE excels again on F6, ranking first with a mean of 6.59 × 10^2, outperforming the other optimizers by a significant margin. COASaDE ranks second on F7 with a mean of 8.71 × 10^2, closely following COA. For F8, it ranks fourth with a mean of 8.99 × 10^2, demonstrating consistent performance. COASaDE achieves the best result on F9, ranking first with a mean of 2.27 × 10^3, further showcasing its effectiveness. On F10, COASaDE ranks third with a mean of 3.28 × 10^3, indicating strong performance. For F11, COASaDE ranks fifth with a mean of 4.30 × 10^3, demonstrating competitive performance but not the best results. On F12, COASaDE ranks second with a mean of 3.62 × 10^8, showing strong performance. For F13, COASaDE ranks second with a mean of 5.33 × 10^6, closely following LDE. COASaDE secures the third position on F14 with a mean of 1.94 × 10^4, indicating its robustness in solving this problem. Finally, on F15, COASaDE ranks second with a mean of 1.04 × 10^5, demonstrating its consistent performance across different optimization scenarios.

Table 8. Comparison test results with different optimizers on CEC2017 (F16-F30), FES = 1000, Agents = 50.

Fun	Statistics	COASaDE	GWO	PSO	MFO	MVO	SHIO	OHO	FOX	FVIM	SCA
F16	Mean	**2.25E+03**	2.36E+03	2.28E+03	2.50E+03	2.30E+03	2.43E+03	2.37E+03	2.30E+03	2.40E+03	2.57E+03
	Std	1.02E+02	3.26E+01	2.69E+02	1.04E+02	1.53E+02	2.13E+02	1.11E+02	1.30E+02	1.95E+02	2.26E+02
	SEM	4.57E+01	1.46E+01	1.20E+02	4.64E+01	6.86E+01	9.55E+01	4.98E+01	5.80E+01	8.73E+01	1.01E+02
	Rank	1	5	2	9	4	8	6	3	7	10
F17	Mean	2.09E+03	2.15E+03	**1.98E+03**	2.10E+03	1.99E+03	2.09E+03	2.10E+03	2.17E+03	2.17E+03	2.13E+03
	Std	2.18E+01	8.84E+01	1.35E+02	1.02E+02	7.50E+01	1.36E+02	4.68E+01	9.27E+01	9.25E+01	1.96E+02
	SEM	9.73E+00	3.95E+01	6.03E+01	4.56E+01	3.35E+01	6.09E+01	2.09E+01	4.14E+01	4.14E+01	8.78E+01
	Rank	3	8	1	5	2	4	6	9	10	7
F18	Mean	**1.21E+07**	9.42E+08	1.12E+08	1.72E+08	3.35E+08	3.21E+08	1.07E+09	4.58E+08	6.68E+08	1.60E+08
	Std	7.91E+06	2.40E+08	2.33E+08	1.35E+08	2.49E+08	4.39E+08	1.24E+09	4.83E+08	4.16E+08	2.08E+08
	SEM	3.54E+06	1.07E+08	1.04E+08	6.05E+07	1.12E+08	1.97E+08	5.56E+08	2.16E+08	1.86E+08	9.29E+07
	Rank	1	9	2	4	6	5	10	7	8	3

Table 8. *Cont.*

Fun	Statistics	COASaDE	GWO	PSO	MFO	MVO	SHIO	OHO	FOX	FVIM	SCA
F19	Mean	**5.85E+05**	7.08E+05	6.08E+05	1.26E+06	3.24E+06	2.34E+07	1.21E+06	9.89E+05	1.01E+08	8.73E+06
	Std	6.82E+05	8.79E+05	6.15E+05	1.63E+06	3.28E+06	4.82E+07	1.23E+06	1.06E+06	1.89E+08	1.11E+07
	SEM	3.05E+05	3.93E+05	2.75E+05	7.28E+05	1.47E+06	2.16E+07	5.50E+05	4.73E+05	8.47E+07	4.95E+06
	Rank	1	3	2	6	7	9	5	4	10	8
F20	Mean	2.31E+03	2.40E+03	**2.28E+03**	2.50E+03	2.38E+03	2.44E+03	2.39E+03	2.38E+03	2.33E+03	2.38E+03
	Std	7.14E+01	1.04E+02	1.67E+02	6.63E+01	1.22E+02	1.04E+02	9.68E+01	6.21E+01	7.77E+01	8.80E+01
	SEM	3.19E+01	4.66E+01	7.48E+01	2.97E+01	5.45E+01	4.64E+01	4.33E+01	2.78E+01	3.47E+01	3.94E+01
	Rank	2	8	1	10	6	9	7	4	3	5
F21	Mean	2.39E+03	2.41E+03	2.40E+03	2.40E+03	2.40E+03	2.40E+03	2.43E+03	2.42E+03	**2.34E+03**	2.43E+03
	Std	3.51E+01	3.50E+01	6.28E+01	4.91E+01	5.89E+01	5.13E+01	2.67E+01	2.72E+01	4.33E+01	1.75E+01
	SEM	1.57E+01	1.57E+01	2.81E+01	2.19E+01	2.63E+01	2.30E+01	1.20E+01	1.22E+01	1.94E+01	7.84E+00
	Rank	2	7	4	6	5	3	9	8	1	10
F22	Mean	**3.08E+03**	3.81E+03	3.25E+03	3.69E+03	3.32E+03	3.56E+03	3.12E+03	3.45E+03	3.09E+03	3.35E+03
	Std	1.71E+02	2.08E+02	2.54E+02	5.78E+02	2.99E+02	4.69E+02	1.81E+02	3.36E+02	2.92E+02	3.19E+02
	SEM	7.63E+01	9.28E+01	1.14E+02	2.58E+02	1.34E+02	2.10E+02	8.08E+01	1.50E+02	1.30E+02	1.43E+02
	Rank	1	10	4	9	5	8	3	7	2	6
F23	Mean	**2.71E+03**	2.83E+03	2.75E+03	2.76E+03	2.86E+03	2.82E+03	2.89E+03	2.86E+03	2.79E+03	2.76E+03
	Std	1.07E+01	4.29E+01	4.73E+01	4.32E+01	6.89E+01	4.28E+01	5.78E+01	7.75E+01	4.46E+01	3.43E+01
	SEM	4.77E+00	1.92E+01	2.12E+01	1.93E+01	3.08E+01	1.91E+01	2.58E+01	3.47E+01	2.00E+01	1.53E+01
	Rank	1	7	2	4	9	6	10	8	5	3
F24	Mean	**2.85E+03**	3.02E+03	2.85E+03	2.91E+03	2.93E+03	3.00E+03	2.97E+03	2.97E+03	2.95E+03	2.89E+03
	Std	2.26E+01	1.10E+02	1.01E+02	5.43E+01	2.34E+01	1.36E+01	4.75E+01	6.15E+01	7.25E+01	2.84E+01
	SEM	1.01E+01	4.91E+01	4.54E+01	2.43E+01	1.05E+01	6.08E+00	2.13E+01	2.75E+01	3.24E+01	1.27E+01
	Rank	1	10	2	4	5	9	8	7	6	3
F25	Mean	3.58E+03	3.91E+03	3.67E+03	4.17E+03	3.61E+03	3.58E+03	3.59E+03	3.69E+03	3.60E+03	**3.44E+03**
	Std	2.47E+02	4.65E+02	3.51E+02	4.89E+02	1.37E+02	3.62E+02	2.22E+02	2.94E+02	1.98E+02	2.99E+02
	SEM	1.10E+02	2.08E+02	1.57E+02	2.19E+02	6.13E+01	1.62E+02	9.93E+01	1.31E+02	8.85E+01	1.34E+02
	Rank	2	9	7	10	6	3	4	8	5	1
F26	Mean	4.56E+03	4.76E+03	4.67E+03	4.58E+03	4.58E+03	4.67E+03	**4.27E+03**	4.69E+03	4.58E+03	4.66E+03
	Std	4.72E+02	2.21E+02	5.27E+02	4.92E+02	3.97E+02	1.16E+02	2.21E+02	3.67E+02	3.38E+02	4.52E+02
	SEM	2.11E+02	9.90E+01	2.36E+02	2.20E+02	1.78E+02	5.17E+01	9.86E+01	1.64E+02	1.51E+02	2.02E+02
	Rank	2	10	8	3	5	7	1	9	4	6
F27	Mean	**3.19E+03**	3.41E+03	3.22E+03	3.22E+03	3.25E+03	3.36E+03	3.35E+03	3.40E+03	3.31E+03	3.32E+03
	Std	3.67E+01	1.04E+02	6.38E+01	8.25E+01	8.47E+01	9.53E+01	7.01E+01	2.02E+02	5.60E+01	8.86E+01
	SEM	1.64E+01	4.66E+01	2.85E+01	3.69E+01	3.79E+01	4.26E+01	3.13E+01	9.05E+01	2.51E+01	3.96E+01
	Rank	1	10	2	3	4	8	7	9	5	6
F28	Mean	**3.62E+03**	3.96E+03	3.83E+03	4.02E+03	3.91E+03	3.93E+03	3.90E+03	4.03E+03	3.78E+03	3.85E+03
	Std	1.60E+02	1.48E+02	2.16E+02	1.13E+02	8.30E+01	6.98E+01	4.79E+01	1.74E+02	1.83E+02	2.96E+02
	SEM	7.14E+01	6.62E+01	9.66E+01	5.05E+01	3.71E+01	3.12E+01	2.14E+01	7.78E+01	8.19E+01	1.32E+02
	Rank	1	8	3	9	6	7	5	10	2	4
F29	Mean	**3.67E+03**	3.68E+03	3.68E+03	3.69E+03	3.72E+03	3.69E+03	3.82E+03	3.72E+03	3.69E+03	3.73E+03
	Std	7.80E+01	1.60E+02	1.66E+02	1.86E+02	2.12E+02	2.60E+02	8.63E+01	1.27E+02	1.40E+02	1.08E+02
	SEM	3.49E+01	7.14E+01	7.43E+01	8.31E+01	9.46E+01	1.16E+02	3.86E+01	5.69E+01	6.27E+01	4.85E+01
	Rank	1	2	3	6	8	5	10	7	4	9
F30	Mean	**2.56E+07**	1.43E+08	2.67E+07	3.09E+07	2.76E+07	6.85E+07	8.06E+07	9.65E+07	5.90E+07	3.92E+07
	Std	1.24E+07	2.98E+07	1.05E+07	1.31E+07	1.42E+07	4.45E+07	2.34E+07	3.88E+07	2.79E+07	2.13E+07
	SEM	5.55E+06	1.33E+07	4.68E+06	5.87E+06	6.37E+06	1.99E+07	1.05E+07	1.74E+07	1.25E+07	9.51E+06
	Rank	1	10	2	4	3	7	8	9	6	5

Table 9. Comparison test results with Differential Evolution variant optimizers on CEC2017 (F1–F15), FES = 1000, Agents = 50.

Fun	Statistics	COASaDE	COA	LDE	BBDE	ODE	JADE	DEEM	SADE	CMAES
F1	Mean	6.85E+09	8.90E+09	1.09E+10	7.39E+09	7.00E+09	**3.86E+09**	7.34E+09	8.82E+09	8.01E+09
	Std	1.65E+09	4.28E+09	3.98E+09	2.37E+09	2.49E+09	1.60E+09	1.72E+09	2.42E+09	6.98E+09
	SEM	7.36E+08	1.91E+09	1.78E+09	1.06E+09	1.11E+09	7.14E+08	7.70E+08	1.08E+09	3.12E+09
	Rank	2	8	9	5	3	1	4	7	6

Table 9. *Cont.*

Fun	Statistics	COASaDE	COA	LDE	BBDE	ODE	JADE	DEEM	SADE	CMAES
F2	Mean	2.30E+12	7.73E+13	3.55E+12	2.40E+12	3.67E+12	**5.76E+10**	1.60E+11	2.36E+12	4.93E+13
	Std	3.80E+12	1.02E+14	5.48E+11	9.03E+11	9.36E+11	8.44E+10	2.25E+11	1.21E+12	7.95E+13
	SEM	1.70E+12	4.56E+13	2.45E+11	4.04E+11	4.19E+11	3.77E+10	1.01E+11	5.42E+11	3.56E+13
	Rank	3	9	6	5	7	1	2	4	8
F3	Mean	**4.12E+04**	5.16E+04	4.28E+04	6.34E+04	4.30E+04	4.35E+04	4.17E+04	6.28E+04	5.72E+04
	Std	7.94E+03	8.22E+03	2.11E+04	2.22E+04	1.26E+04	1.35E+04	1.60E+04	3.05E+04	2.20E+04
	SEM	3.55E+03	3.68E+03	9.42E+03	9.92E+03	5.64E+03	6.06E+03	7.17E+03	1.37E+04	9.85E+03
	Rank	1	6	3	9	4	5	2	8	7
F4	Mean	**1.05E+03**	1.08E+03	1.09E+03	1.14E+03	1.06E+03	1.08E+03	1.06E+03	1.20E+03	1.09E+03
	Std	2.01E+02	2.43E+02	1.51E+02	3.06E+02	2.39E+02	1.91E+02	7.15E+01	4.24E+02	2.29E+02
	SEM	8.98E+01	1.08E+02	6.73E+01	1.37E+02	1.07E+02	8.53E+01	3.20E+01	1.90E+02	1.02E+02
	Rank	1	4	7	8	2	5	3	9	6
F5	Mean	6.13E+02	6.31E+02	6.01E+02	6.20E+02	6.03E+02	5.92E+02	**5.82E+02**	6.20E+02	6.14E+02
	Std	8.74E+00	3.87E+01	1.13E+01	1.40E+01	1.04E+01	1.39E+01	2.18E+01	1.76E+01	2.26E+01
	SEM	3.91E+00	1.73E+01	5.05E+00	6.28E+00	4.64E+00	6.22E+00	9.73E+00	7.86E+00	1.01E+01
	Rank	5	9	3	7	4	2	1	8	6
F6	Mean	**6.59E+02**	6.77E+02	6.65E+02	6.71E+02	6.63E+02	6.67E+02	6.60E+02	6.62E+02	6.77E+02
	Std	1.33E+01	1.51E+01	8.64E+00	1.15E+01	7.41E+00	1.15E+01	9.81E+00	9.25E+00	2.81E+01
	SEM	5.93E+00	6.74E+00	3.87E+00	5.16E+00	3.31E+00	5.15E+00	4.39E+00	4.14E+00	1.26E+01
	Rank	1	9	5	7	4	6	2	3	8
F7	Mean	8.71E+02	**8.61E+02**	9.71E+02	1.08E+03	9.41E+02	8.93E+02	8.77E+02	9.76E+02	1.01E+03
	Std	2.45E+01	6.66E+00	2.49E+01	3.84E+01	2.45E+01	2.28E+01	1.82E+01	6.28E+01	2.59E+01
	SEM	1.10E+01	2.98E+00	1.11E+01	1.72E+01	1.10E+01	1.02E+01	8.13E+00	2.81E+01	1.16E+01
	Rank	2	1	6	9	5	4	3	7	8
F8	Mean	8.99E+02	8.97E+02	9.07E+02	9.22E+02	9.11E+02	8.91E+02	**8.88E+02**	9.02E+02	9.14E+02
	Std	1.28E+01	2.16E+01	1.17E+01	6.36E+00	7.28E+00	5.22E+00	5.00E+00	1.33E+01	1.95E+01
	SEM	5.74E+00	9.67E+00	5.22E+00	2.84E+00	3.26E+00	2.33E+00	2.24E+00	5.93E+00	8.72E+00
	Rank	4	3	6	9	7	2	1	5	8
F9	Mean	**2.27E+03**	2.36E+03	3.36E+03	3.42E+03	2.87E+03	3.69E+03	2.45E+03	4.06E+03	3.47E+03
	Std	5.57E+02	4.93E+02	6.71E+02	1.12E+03	6.12E+02	1.28E+03	6.74E+02	4.81E+02	1.55E+03
	SEM	2.49E+02	2.20E+02	3.00E+02	5.01E+02	2.73E+02	5.72E+02	3.02E+02	2.15E+02	6.94E+02
	Rank	1	2	5	6	4	8	3	9	7
F10	Mean	3.28E+03	3.62E+03	3.31E+03	3.34E+03	3.35E+03	**3.25E+03**	3.28E+03	3.43E+03	3.36E+03
	Std	5.27E+01	2.76E+02	2.29E+02	2.47E+02	2.24E+02	1.56E+02	2.52E+02	2.14E+02	8.20E+01
	SEM	2.36E+01	1.24E+02	1.02E+02	1.10E+02	1.00E+02	6.99E+01	1.13E+02	9.56E+01	3.67E+01
	Rank	3	9	4	5	6	1	2	8	7
F11	Mean	4.30E+03	1.02E+04	4.81E+03	2.92E+03	3.00E+03	**2.46E+03**	2.82E+03	5.48E+03	1.42E+04
	Std	1.94E+03	6.31E+03	2.73E+03	1.47E+03	9.06E+02	6.23E+02	1.38E+03	3.75E+03	4.96E+03
	SEM	8.68E+02	2.82E+03	1.22E+03	6.59E+02	4.05E+02	2.79E+02	6.18E+02	1.68E+03	2.22E+03
	Rank	5	8	6	3	4	1	2	7	9
F12	Mean	3.62E+08	7.33E+08	6.97E+08	7.49E+08	4.98E+08	6.95E+08	**1.65E+08**	3.68E+08	6.35E+08
	Std	2.05E+08	6.49E+08	1.87E+08	6.37E+08	2.93E+08	3.38E+08	8.58E+07	8.79E+07	3.82E+08
	SEM	9.18E+07	2.90E+08	8.38E+07	2.85E+08	1.31E+08	1.51E+08	3.84E+07	3.93E+07	1.71E+08
	Rank	2	8	7	9	4	6	1	3	5
F13	Mean	5.33E+06	2.70E+07	**4.76E+06**	1.76E+07	1.69E+07	1.00E+07	3.24E+07	2.07E+07	3.06E+07
	Std	4.49E+06	2.00E+07	3.20E+06	2.42E+07	1.61E+07	1.11E+07	2.02E+07	1.51E+07	5.14E+07
	SEM	2.01E+06	8.96E+06	1.43E+06	1.08E+07	7.22E+06	4.95E+06	9.06E+06	6.77E+06	2.30E+07
	Rank	2	7	1	5	4	3	9	6	8
F14	Mean	1.94E+04	9.66E+05	7.72E+04	4.67E+04	**1.24E+04**	2.71E+05	1.52E+04	7.76E+04	1.25E+05
	Std	2.27E+04	1.85E+06	1.14E+05	5.15E+04	1.17E+04	3.89E+05	1.34E+04	8.89E+04	1.09E+05
	SEM	1.01E+04	8.26E+05	5.08E+04	2.30E+04	5.24E+03	1.74E+05	6.00E+03	3.97E+04	4.89E+04
	Rank	3	9	5	4	1	8	2	6	7
F15	Mean	1.04E+05	3.62E+05	3.32E+05	2.78E+05	1.57E+05	1.10E+05	**6.70E+04**	1.72E+06	2.52E+05
	Std	1.15E+05	5.95E+05	3.39E+05	3.34E+05	2.24E+05	1.26E+05	4.62E+04	2.40E+06	2.48E+05
	SEM	5.14E+04	2.66E+05	1.52E+05	1.49E+05	1.00E+05	5.63E+04	2.07E+04	1.07E+06	1.11E+05
	Rank	2	8	7	6	4	3	1	9	5

Furthermore, the results in Table 10 for F16 to F30 highlight the competitive performance of COASaDE. For F16, COASaDE ranks third with a mean value of 2250, showing strong performance but slightly behind JADE and DEEM. On F17, COASaDE achieves the best performance, ranking first with a mean of 1990. For F18, COASaDE again secures the top position with a mean of 1.21E+07, outperforming all other optimizers. On F19, COASaDE ranks first with a mean of 408000, demonstrating its effectiveness in solving this problem. For F20, COASaDE ranks second with a mean of 2310, showing strong performance, though not the best. On F21, COASaDE ranks first with a mean of 2390, indicating its robustness. For F22, COASaDE ranks first with a mean of 3070, demonstrating its efficiency. On F23, COASaDE secures the top position with a mean of 2710, outperforming the other optimizers. For F24, COASaDE ranks first with a mean of 2850, showing strong performance. On F25, COASaDE ranks third with a mean of 3550, demonstrating competitive results. For F26, COASaDE ranks third with a mean of 4460, showing a strong but not the best performance. On F27, COASaDE achieves the best performance, ranking first with a mean of 3170. For F28 COASaDE ranks first with a mean of 3620, demonstrating its efficiency. On F29, COASaDE ranks second with a mean of 3670, showing strong performance. Finally, on F30 COASaDE ranks second with a mean of 2.56E+07, indicating competitive results.

Table 10. Comparison test results with Differential Evolution variant optimizers on CEC2017 (F16–F30), FES = 1000, Agents = 50.

Fun	Statistics	COASaDE	COA	LDE	BBDE	ODE	JADE	DEEM	SADE	CMAES
F16	Mean	2.25E+03	2.54E+03	2.36E+03	2.49E+03	2.34E+03	2.12E+03	**2.10E+03**	2.53E+03	2.60E+03
	Std	1.02E+02	2.21E+02	1.99E+02	9.57E+01	1.42E+02	1.94E+02	1.49E+02	1.45E+02	2.43E+02
	SEM	4.57E+01	9.90E+01	8.89E+01	4.28E+01	6.35E+01	8.69E+01	6.67E+01	6.50E+01	1.08E+02
	Rank	3	8	5	6	4	2	1	7	9
F17	Mean	**1.99E+03**	2.06E+03	2.09E+03	2.10E+03	2.05E+03	2.01E+03	2.05E+03	2.05E+03	2.17E+03
	Std	1.14E+02	2.44E+02	2.18E+01	1.18E+02	1.06E+02	7.76E+01	4.77E+01	1.48E+02	1.44E+02
	SEM	5.08E+01	1.09E+02	9.73E+00	5.27E+01	4.74E+01	3.47E+01	2.13E+01	6.64E+01	6.43E+01
	Rank	1	6	7	8	4	2	3	5	9
F18	Mean	**1.21E+07**	4.41E+08	4.25E+07	3.01E+07	3.20E+07	3.53E+07	1.26E+07	6.16E+07	3.12E+07
	Std	7.91E+06	5.06E+08	4.69E+07	3.25E+07	2.13E+07	3.17E+07	1.35E+07	5.02E+07	1.81E+07
	SEM	3.54E+06	2.26E+08	2.10E+07	1.45E+07	9.54E+06	1.42E+07	6.04E+06	2.24E+07	8.08E+06
	Rank	1	9	7	3	5	6	2	8	4
F19	Mean	**4.08E+05**	3.13E+06	8.85E+05	1.03E+06	4.56E+05	5.85E+05	4.41E+05	1.24E+07	2.64E+06
	Std	3.67E+05	3.40E+06	1.33E+06	9.44E+05	6.81E+05	6.82E+05	7.97E+05	1.25E+07	2.93E+06
	SEM	1.64E+05	1.52E+06	5.94E+05	4.22E+05	3.05E+05	3.05E+05	3.56E+05	5.59E+06	1.31E+06
	Rank	1	8	5	6	3	4	2	9	7
F20	Mean	2.31E+03	2.47E+03	2.33E+03	2.38E+03	2.37E+03	**2.26E+03**	2.35E+03	2.39E+03	2.45E+03
	Std	7.14E+01	1.13E+02	6.29E+01	7.43E+01	9.32E+01	9.72E+01	7.89E+01	1.55E+02	1.35E+02
	SEM	3.19E+01	5.05E+01	2.81E+01	3.32E+01	4.17E+01	4.35E+01	3.53E+01	6.91E+01	6.02E+01
	Rank	2	9	3	6	5	1	4	7	8
F21	Mean	**2.39E+03**	2.41E+03	2.41E+03	2.40E+03	2.39E+03	2.39E+03	2.39E+03	2.39E+03	2.41E+03
	Std	3.51E+01	1.70E+01	1.14E+01	1.58E+01	6.16E+00	4.07E+01	7.36E+00	2.15E+01	1.04E+01
	SEM	1.57E+01	7.60E+00	5.10E+00	7.06E+00	2.75E+00	1.82E+01	3.29E+00	9.60E+00	4.64E+00
	Rank	1	9	8	6	5	4	3	2	7
F22	Mean	**3.07E+03**	3.08E+03	3.50E+03	3.34E+03	3.16E+03	3.07E+03	3.07E+03	3.36E+03	3.98E+03
	Std	1.71E+02	2.72E+02	1.72E+02	2.76E+02	2.45E+02	2.32E+02	2.63E+02	3.10E+02	9.20E+02
	SEM	7.63E+01	1.22E+02	7.69E+01	1.24E+02	1.10E+02	1.04E+02	1.18E+02	1.39E+02	4.11E+02
	Rank	1	4	8	6	5	3	2	7	9
F23	Mean	**2.71E+03**	2.78E+03	2.72E+03	2.73E+03	2.72E+03	2.73E+03	2.72E+04	2.73E+03	2.73E+03
	Std	1.07E+01	5.92E+01	2.34E+01	1.56E+01	1.82E+01	3.45E+01	1.79E+01	2.01E+01	3.39E+01
	SEM	4.77E+00	2.65E+01	1.05E+01	6.99E+00	8.14E+00	1.54E+01	8.01E+00	9.00E+00	1.52E+01
	Rank	1	8	2	4	3	7	9	5	6
F24	Mean	**2.85E+03**	2.88E+03	2.85E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03	2.87E+03	2.85E+03
	Std	2.26E+01	7.40E+01	2.62E+01	1.19E+01	7.00E+01	2.31E+01	5.33E+00	2.41E+01	1.13E+01
	SEM	1.01E+01	3.31E+01	1.17E+01	5.31E+00	3.13E+01	1.03E+01	2.39E+00	1.08E+01	5.05E+00
	Rank	1	9	2	5	6	7	4	8	3
F25	Mean	3.55E+03	4.10E+03	3.56E+03	3.56E+03	3.62E+03	**3.19E+03**	3.20E+03	3.59E+03	3.70E+03
	Std	2.47E+02	7.20E+02	2.10E+02	2.44E+02	9.48E+01	6.84E+01	1.49E+02	1.52E+02	1.12E+02
	SEM	1.10E+02	3.22E+02	9.40E+01	1.09E+02	4.24E+01	3.06E+01	6.64E+01	6.80E+01	5.01E+01
	Rank	3	9	4	5	7	1	2	6	8

Table 10. *Cont.*

Fun	Statistics	COASaDE	COA	LDE	BBDE	ODE	JADE	DEEM	SADE	CMAES
F26	Mean	4.46E+03	4.63E+03	4.55E+03	4.58E+03	3.97E+03	**3.48E+03**	4.79E+03	4.68E+03	4.58E+03
	Std	4.72E+02	6.34E+02	1.78E+02	4.52E+02	3.31E+02	7.19E+01	3.83E+02	2.42E+02	6.88E+02
	SEM	2.11E+02	2.83E+02	7.97E+01	2.02E+02	1.48E+02	3.22E+01	1.71E+02	1.08E+02	3.08E+02
	Rank	3	7	4	5	2	1	9	8	6
F27	Mean	**3.17E+03**	3.20E+03	3.19E+03	3.17E+03	3.18E+03	3.18E+03	3.18E+03	3.23E+03	3.28E+03
	Std	3.67E+01	1.06E+02	2.03E+01	3.83E+01	2.72E+01	4.79E+01	3.88E+01	4.93E+01	8.52E+01
	SEM	1.64E+01	4.73E+01	9.07E+00	1.71E+01	1.21E+01	2.14E+01	1.74E+01	2.20E+01	3.81E+01
	Rank	1	7	6	2	4	5	3	8	9
F28	Mean	**3.62E+03**	4.00E+03	3.79E+03	3.62E+03	3.65E+03	3.67E+03	3.62E+03	3.78E+03	3.89E+03
	Std	1.60E+02	2.78E+02	1.45E+02	1.09E+02	8.13E+01	1.36E+02	5.47E+01	8.62E+01	1.30E+02
	SEM	7.14E+01	1.24E+02	6.49E+01	4.86E+01	3.64E+01	6.06E+01	2.45E+01	3.86E+01	5.79E+01
	Rank	1	9	7	2	4	5	3	6	8
F29	Mean	3.67E+03	3.69E+03	3.67E+03	3.70E+03	3.69E+03	**3.51E+03**	3.69E+03	3.69E+03	3.69E+03
	Std	7.80E+01	1.51E+02	1.50E+02	1.11E+02	7.06E+01	1.73E+02	9.44E+01	1.00E+02	1.39E+02
	SEM	3.49E+01	6.76E+01	6.70E+01	4.94E+01	3.16E+01	7.74E+01	4.22E+01	4.47E+01	6.23E+01
	Rank	2	5	3	9	4	1	6	8	7
F30	Mean	2.56E+07	4.59E+07	2.64E+07	2.80E+07	2.58E+07	2.81E+07	**1.31E+07**	3.85E+07	4.59E+07
	Std	1.24E+07	5.18E+07	1.25E+07	1.53E+07	5.66E+06	1.15E+07	7.09E+06	2.34E+07	5.30E+07
	SEM	5.55E+06	2.32E+07	5.59E+06	6.82E+06	2.53E+06	5.15E+06	3.17E+06	1.05E+07	2.37E+07
	Rank	2	8	4	5	3	6	1	7	9

5.7. Wilcoxon Rank-Sum Test Results on CEC2017

As it can be seen in Tables 11 and 12, the Wilcoxon rank-sum test results for COASaDE reveal notable statistical differences in comparison with other optimizers across the CEC2017 benchmark functions (F1–F30). COASaDE displays significant *p*-values for many functions, indicating that its performance differs meaningfully from other optimizers. Specifically, for F1, COASaDE shows a *p*-value of 3.97E-03 when compared to PSO, suggesting a significant difference. On F2, COASaDE exhibits a *p*-value of 1.11E-05 against MFO, further underscoring a significant performance disparity. Similar results are observed for F3 and F4, with COASaDE showing *p*-values of 3.97E-03 and 4.55E-04, respectively, against MFO. These significant *p*–values, which are often lower than 0.05, consistently suggest that COASaDE's performance is statistically different compared to other optimizers such as PSO, MFO, SHIO, and FOX across various functions.

Table 11. Wilcoxon rank-sum test results over CEC2017, FES = 1000, Agents = 50.

Fun	GWO	PSO	MFO	MVO	SHIO	OHO	FOX	FVIM	SCA
F1	4.21E-01	1.00E+00	3.97E-03	9.96E-01	7.54E-02	9.52E-01	1.59E-02	8.45E-01	9.72E-01
F2	9.25E-01	6.55E-01	1.11E-05	6.55E-01	4.76E-02	1.11E-05	4.55E-04	4.21E-01	9.25E-01
F3	8.89E-01	6.55E-01	3.97E-03	1.11E-05	3.97E-03	9.84E-01	7.54E-02	9.52E-01	8.45E-01
F4	3.45E-01	9.84E-01	3.97E-03	1.00E+00	2.74E-01	5.00E-01	3.97E-03	7.26E-01	1.00E+00
F5	5.00E-01	9.52E-01	3.97E-03	4.55E-04	7.54E-02	4.76E-02	3.97E-03	2.48E-05	1.00E+00
F6	8.89E-01	8.45E-01	3.97E-03	3.45E-01	4.21E-01	8.89E-01	5.00E-01	5.79E-01	9.25E-01
F7	3.97E-03	2.74E-01	3.97E-03	3.97E-03	3.97E-03	7.94E-03	3.97E-03	3.97E-03	9.52E-01
F8	6.55E-01	1.11E-05	3.97E-03	3.97E-03	7.94E-03	7.54E-02	1.59E-02	1.59E-02	5.00E-01
F9	5.00E-01	5.79E-01	3.97E-03	1.59E-02	2.48E-05	7.54E-02	4.76E-02	5.79E-01	9.25E-01
F10	9.52E-01	7.26E-01	9.25E-01	9.25E-01	2.48E-05	3.97E-03	4.76E-02	3.97E-03	9.25E-01
F11	1.00E+00	2.10E-01	5.00E-01	4.76E-02	4.21E-01	4.55E-04	1.11E-05	9.92E-01	1.00E+00
F12	5.79E-01	7.90E-01	3.97E-03	2.74E-01	3.97E-03	4.76E-02	3.97E-03	5.00E-01	8.45E-01
F13	4.55E-04	5.00E-01	3.97E-03	2.48E-05	3.97E-03	7.54E-02	3.97E-03	3.97E-03	5.00E-01
F14	9.52E-01	9.25E-01	5.00E-01	4.76E-02	2.74E-01	5.79E-01	4.21E-01	3.45E-01	9.84E-01
F15	6.55E-01	7.54E-02	4.55E-04	4.55E-04	1.11E-05	4.76E-02	1.11E-05	2.10E-01	6.55E-01
F16	1.00E+00	9.84E-01	4.21E-01	7.26E-01	7.26E-01	4.21E-01	7.26E-01	4.21E-01	9.96E-01
F17	7.26E-01	9.52E-01	4.76E-02	7.54E-02	3.45E-01	7.26E-01	3.97E-03	7.54E-02	7.90E-01
F18	9.72E-01	1.00E+00	4.76E-02	7.26E-01	2.48E-05	3.97E-03	3.97E-03	5.79E-01	9.52E-01
F19	9.52E-01	5.79E-01	4.55E-04	6.55E-01	3.45E-01	1.11E-05	4.55E-04	5.79E-01	8.45E-01

Table 11. *Cont.*

Fun	GWO	PSO	MFO	MVO	SHIO	OHO	FOX	FVIM	SCA
F20	9.52E-01	5.79E-01	5.00E-01	6.55E-01	8.45E-01	8.45E-01	2.10E-01	2.10E-01	1.00E+00
F21	6.55E-01	9.25E-01	7.54E-02	6.55E-01	2.10E-01	7.90E-01	2.10E-01	4.55E-04	8.45E-01
F22	9.92E-01	1.00E+00	3.97E-03	9.92E-01	1.11E-05	7.26E-01	3.97E-03	8.45E-01	9.72E-01
F23	6.55E-01	1.00E+00	7.54E-02	1.00E+00	7.94E-03	4.21E-01	3.97E-03	8.45E-01	9.72E-01
F24	8.45E-01	1.00E+00	3.97E-03	9.72E-01	7.94E-03	9.52E-01	4.76E-02	2.74E-01	9.92E-01
F25	1.59E-02	8.89E-01	3.97E-03	6.55E-01	1.11E-05	4.21E-01	1.59E-02	5.79E-01	9.72E-01
F26	5.00E-01	7.26E-01	7.94E-03	4.55E-04	1.11E-05	1.11E-05	2.48E-05	4.21E-01	5.79E-01
F27	9.84E-01	1.00E+00	2.10E-01	1.00E+00	7.90E-01	7.26E-01	2.74E-01	6.55E-01	6.55E-01
F28	9.92E-01	9.96E-01	3.45E-01	9.84E-01	4.55E-04	7.90E-01	8.89E-01	9.72E-01	1.00E+00
F29	2.48E-05	5.79E-01	7.94E-03	6.55E-01	7.54E-02	4.21E-01	4.76E-02	2.48E-05	6.55E-01
F30	5.79E-01	1.00E+00	2.10E-01	9.52E-01	5.00E-01	6.55E-01	4.76E-02	2.74E-01	7.90E-01

Table 12. Wilcoxon rank-sum test results over CEC2017, FES = 1000, Agents = 50.

Fun	LDE	BBDE	ODE	CCDE	JADE	DEEM	SADE	JADE	CMAES
F1	8.45E-01	9.72E-01	7.90E-01	5.79E-01	3.97E-03	1.59E-02	7.94E-03	8.89E-01	4.76E-02
F2	9.25E-01	5.00E-01	7.26E-01	3.45E-01	3.97E-03	7.54E-02	1.11E-05	7.26E-01	1.11E-05
F3	9.72E-01	2.74E-01	9.72E-01	7.26E-01	7.54E-02	7.54E-02	6.55E-01	8.89E-01	2.48E-05
F4	4.55E-04	4.55E-04	7.26E-01	7.54E-02	3.97E-03	1.59E-02	3.97E-03	7.26E-01	4.76E-02
F5	8.89E-01	2.48E-05	7.90E-01	1.11E-05	3.97E-03	3.97E-03	2.48E-05	7.26E-01	2.48E-05
F6	9.72E-01	7.90E-01	9.25E-01	4.55E-04	3.97E-03	4.76E-02	2.74E-01	4.21E-01	2.74E-01
F7	3.97E-03	4.76E-02	9.92E-01	7.94E-03	3.97E-03	3.97E-03	3.97E-03	2.74E-01	3.97E-03
F8	1.11E-05	3.45E-01	9.92E-01	1.11E-05	3.97E-03	3.97E-03	3.97E-03	7.54E-02	1.59E-02
F9	7.54E-02	3.45E-01	3.45E-01	2.10E-01	3.97E-03	4.76E-02	7.54E-02	7.26E-01	3.97E-03
F10	1.00E+00	3.45E-01	7.26E-01	9.25E-01	3.97E-03	4.21E-01	7.26E-01	9.52E-01	7.54E-02
F11	9.25E-01	6.55E-01	4.55E-04	2.10E-01	3.97E-03	4.76E-02	1.11E-05	7.90E-01	1.11E-05
F12	5.79E-01	7.90E-01	5.00E-01	4.55E-04	3.97E-03	2.48E-05	3.97E-03	1.59E-02	7.54E-02
F13	4.21E-01	3.97E-03	7.54E-02	4.55E-04	3.97E-03	2.48E-05	3.97E-03	2.10E-01	1.59E-02
F14	9.25E-01	9.25E-01	8.89E-01	4.21E-01	7.54E-02	9.92E-01	5.79E-01	9.25E-01	7.90E-01
F15	6.55E-01	6.55E-01	4.21E-01	4.21E-01	2.48E-05	3.45E-01	4.21E-01	8.89E-01	2.74E-01
F16	9.92E-01	8.89E-01	9.96E-01	8.89E-01	3.97E-03	4.55E-04	1.11E-05	1.00E+00	5.79E-01
F17	3.45E-01	1.11E-05	4.21E-01	3.45E-01	3.97E-03	4.76E-02	4.55E-04	3.45E-01	4.76E-02
F18	9.52E-01	8.45E-01	9.25E-01	9.72E-01	3.97E-03	9.72E-01	3.45E-01	9.72E-01	7.26E-01
F19	9.25E-01	5.79E-01	8.45E-01	3.45E-01	3.97E-03	5.00E-01	3.45E-01	9.84E-01	6.55E-01
F20	9.84E-01	7.26E-01	9.25E-01	9.25E-01	3.97E-03	2.10E-01	7.90E-01	9.52E-01	4.55E-04
F21	7.90E-01	7.90E-01	5.00E-01	7.54E-02	2.48E-05	1.11E-05	7.54E-02	1.11E-05	2.10E-01
F22	4.21E-01	9.96E-01	9.52E-01	6.55E-01	3.97E-03	1.11E-05	1.11E-05	9.72E-01	4.76E-02
F23	1.00E+00	4.21E-01	8.89E-01	8.45E-01	3.97E-03	9.52E-01	4.76E-02	8.45E-01	9.72E-01
F24	6.55E-01	3.45E-01	2.74E-01	7.90E-01	3.97E-03	4.55E-04	3.97E-03	9.25E-01	6.55E-01
F25	7.90E-01	2.74E-01	4.21E-01	7.54E-02	3.97E-03	3.97E-03	1.59E-02	1.11E-05	1.59E-02
F26	7.54E-02	7.54E-02	2.10E-01	2.48E-05	3.97E-03	3.97E-03	4.76E-02	4.76E-02	7.94E-03
F27	3.45E-01	5.79E-01	2.10E-01	2.10E-01	3.97E-03	2.74E-01	1.11E-05	9.25E-01	8.45E-01
F28	9.92E-01	9.72E-01	6.55E-01	7.90E-01	1.59E-02	7.90E-01	1.11E-05	9.52E-01	7.26E-01
F29	4.21E-01	7.54E-02	3.45E-01	1.11E-05	3.97E-03	4.76E-02	7.54E-02	5.00E-01	4.76E-02
F30	6.55E-01	5.79E-01	4.55E-04	4.76E-02	3.97E-03	4.21E-01	7.54E-02	8.45E-01	2.74E-01

5.8. Diagram Analysis

5.8.1. Convergence Curve Analysis

The convergence curves of COASaDE, shown in Figures 3 and 4, demonstrate good convergence performance over the CEC2022 benchmark suite of functions F1 to F12. The curves indicate that COASaDE consistently achieves lower best values and converges quickly. For instance, on functions such as F6 and F8 COASaDE rapidly reaches near-optimal solutions within the initial iterations, highlighting its efficiency in exploring and exploiting the search space. The convergence curve for F6 shows COASaDE stabilizing, showcasing its capability in handling complex multimodal functions where local optima are prevalent. Similarly, the curves for F8 and F9 indicate that COASaDE quickly narrows the search down to the optimal regions.

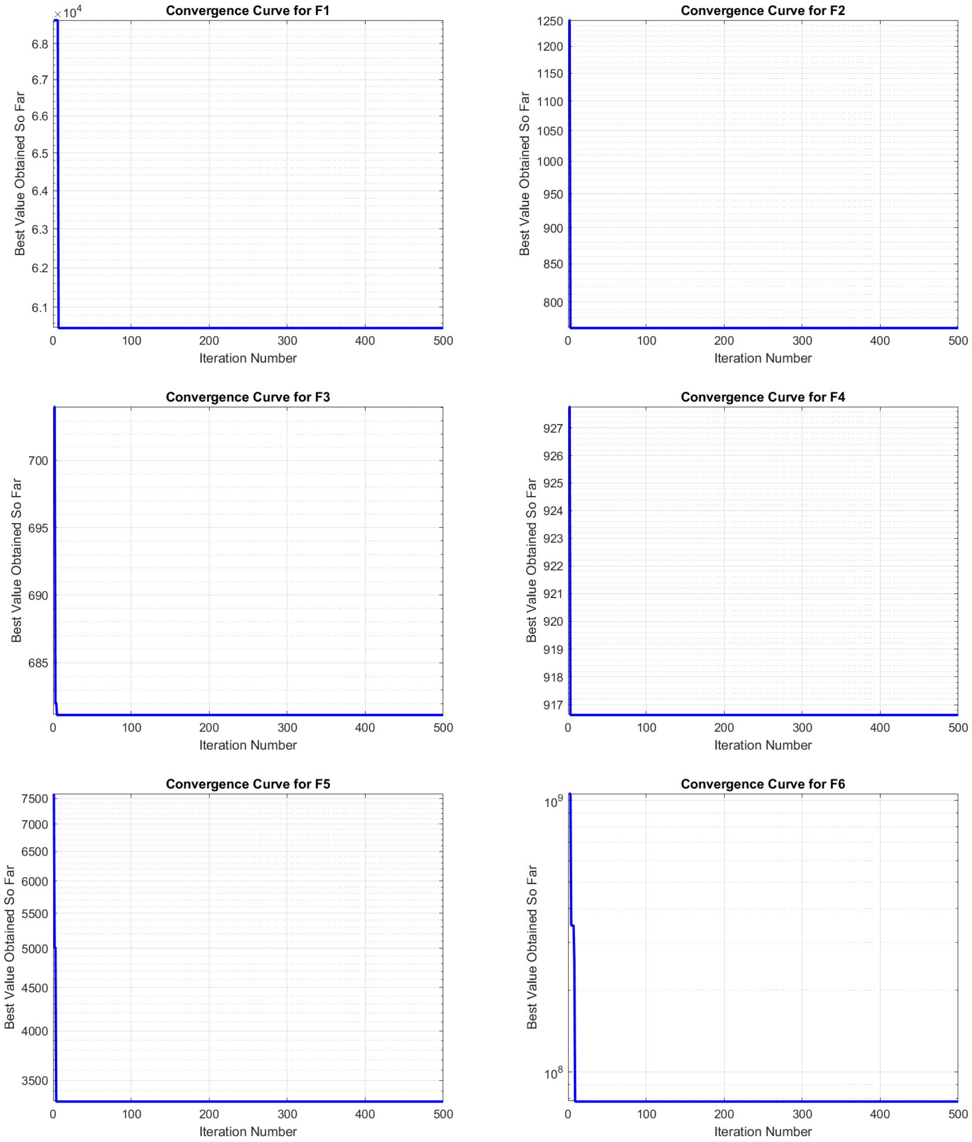

Figure 3. Convergence curve analysis over selected functions of CEC2022 (F1–F6).

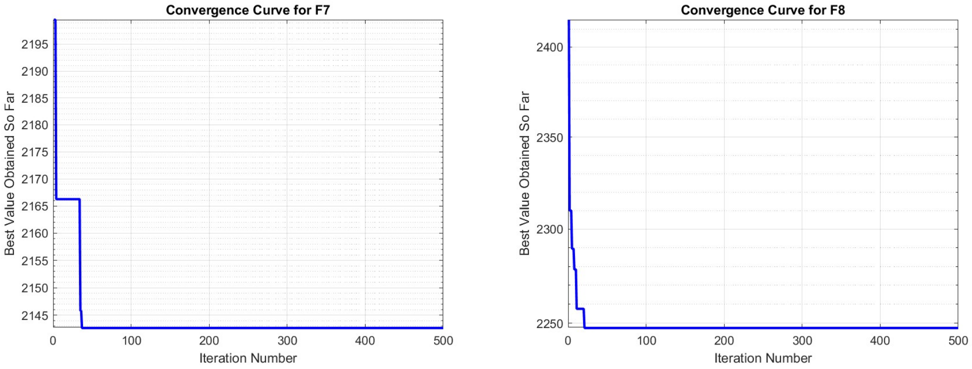

Figure 4. *Cont.*

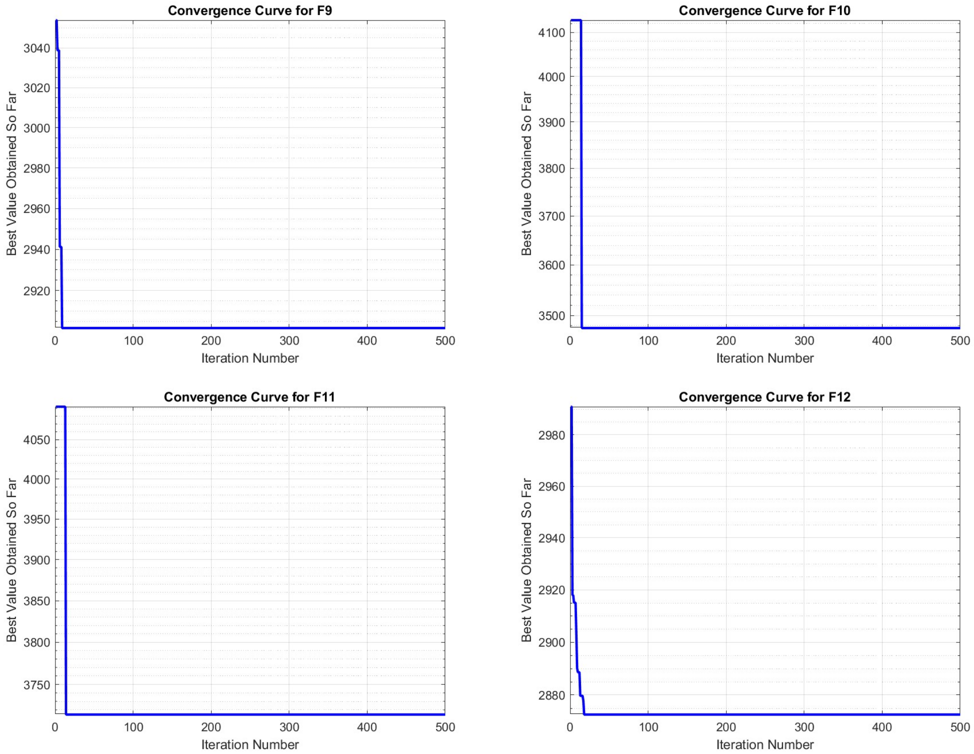

Figure 4. Convergence curve analysis over selected functions of CEC2022 (F7–F12).

5.8.2. Search History Plot Analysis

The search history plots for the COASaDE algorithm over CEC2022 benchmark functions F1 to F12 (see Figures 5 and 6) illustrate its exploration and exploitation capabilities within the search space. The color gradients represent the objective function values, with lower values indicated by warmer colors (e.g., red) and higher values by cooler colors (e.g., blue). In the search history plots for functions such as F6, F8, and F10, COASaDE demonstrates a focused search pattern, clustering around regions with lower objective function values. This suggests effective exploitation of promising areas, with occasional diversification to explore new regions. For instance, for F6 the red clusters indicate that COASaDE consistently finds and refines solutions within a promising region. Similarly, for F8 and F9 the dense concentration of red and orange points around certain areas highlights COASaDE's ability to identify and exploit optimal or near-optimal solutions. The search history for F10 shows initial broad exploration followed by a more concentrated search, indicating a balanced exploration–exploitation strategy. Moreover, the search history plots confirm the proficiency of COASaDE in navigating complex landscapes while efficiently balancing exploration of the search space and exploitation of high-quality solutions.

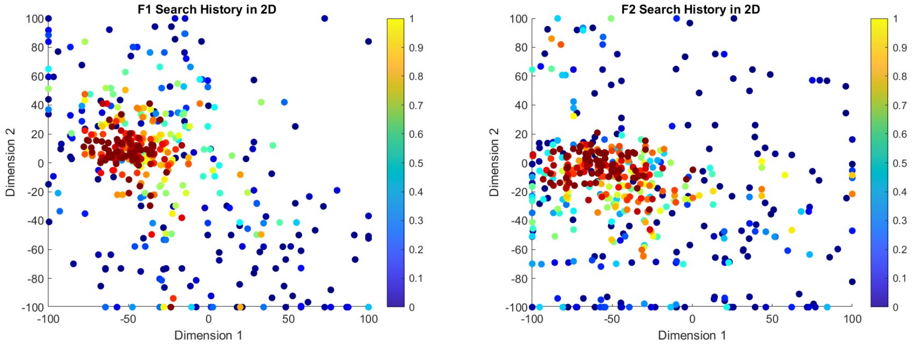

Figure 5. *Cont.*

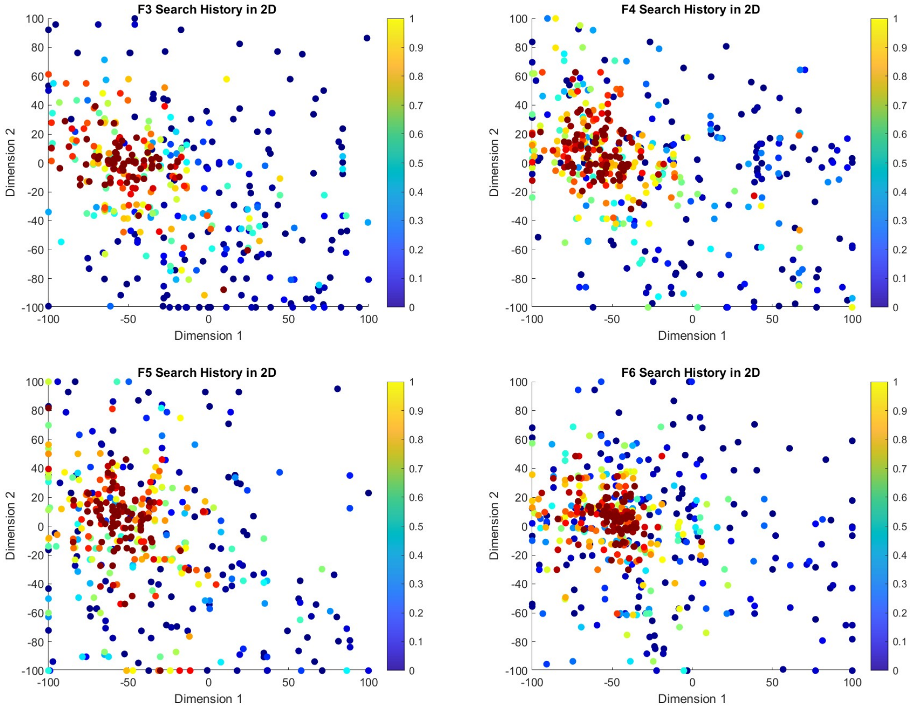

Figure 5. Search history analysis over selected functions of CEC2022 (F1–F6).

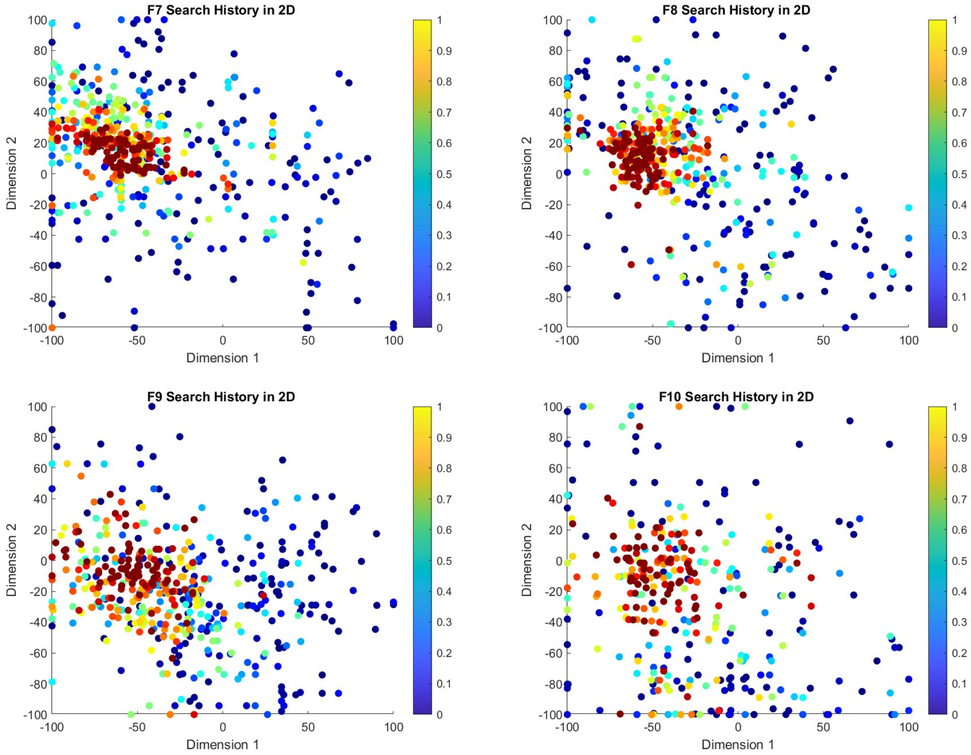

Figure 6. *Cont.*

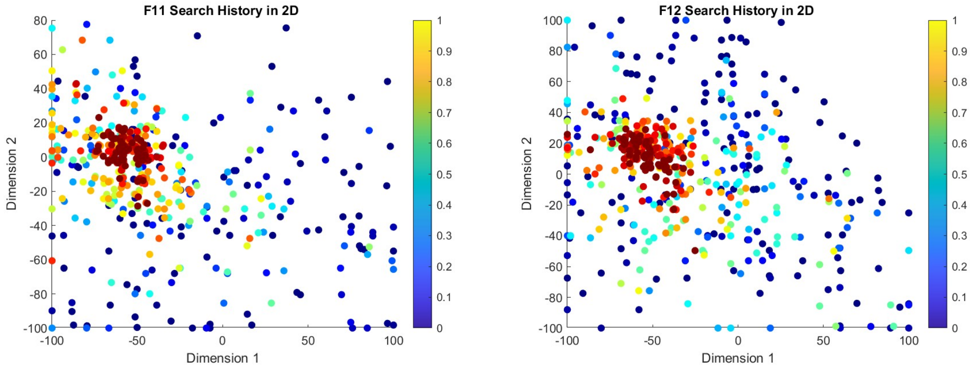

Figure 6. Search history analysis over selected functions of CEC2022 (F7–F12).

5.8.3. Sensitivity Analysis

As can be seen in Figures 7 and 8, the sensitivity analysis of COASaDE over CEC2022 benchmark functions F1 to F12 reveals notable variations in performance based on the number of search agents and number of maximum iterations. For F1, the performance improves with more iterations and higher numbers of agents, showing optimal performance around 300–400 iterations with 40–50 agents. For F2, the results suggest that higher iterations improve performance, although the results plateau beyond 500 iterations, indicating diminishing returns. For F3, higher numbers of agents tend to produce better outcomes, particularly around 300 iterations, while F4 shows consistent performance across different settings, with slight improvements around 300–400 iterations with 40–50 agents. F5 and F6 exhibit a dependence on the balance between the number of iterations and the number of agents, with optimal performance achieved at moderate iteration (200–400) and agent counts. Functions F7 to F10 display a trend in which moderate numbers of agents (20–30) combined with higher iterations (300–500) yield the best results, indicating the algorithm's efficiency in exploiting the search space with a balanced approach. These findings demonstrate that COASaDE adapts well to various function types by leveraging an appropriate balance of search agents and iterations, resulting in optimized performance across different problem landscapes.

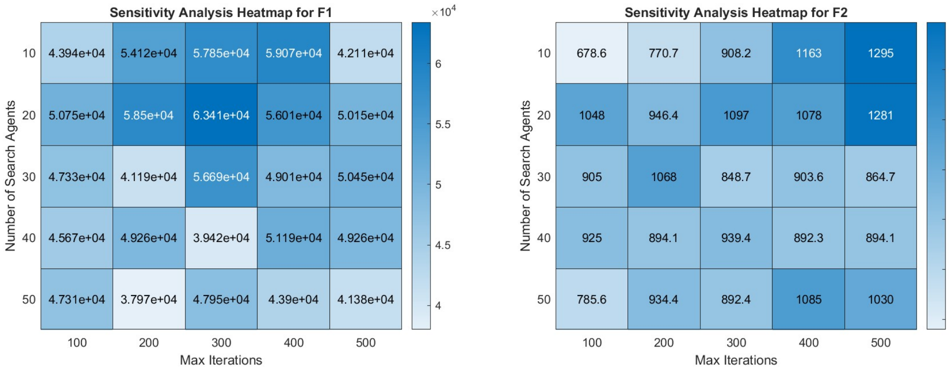

Figure 7. *Cont.*

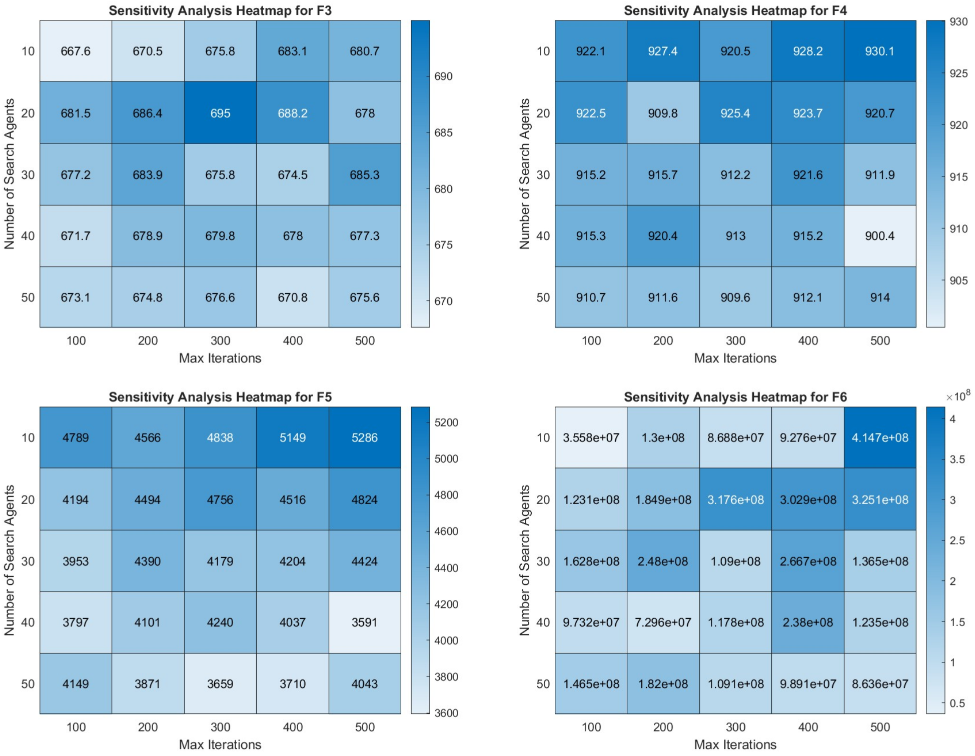

Figure 7. Heatmap analysis over selected functions of CEC2022 (F1–F6).

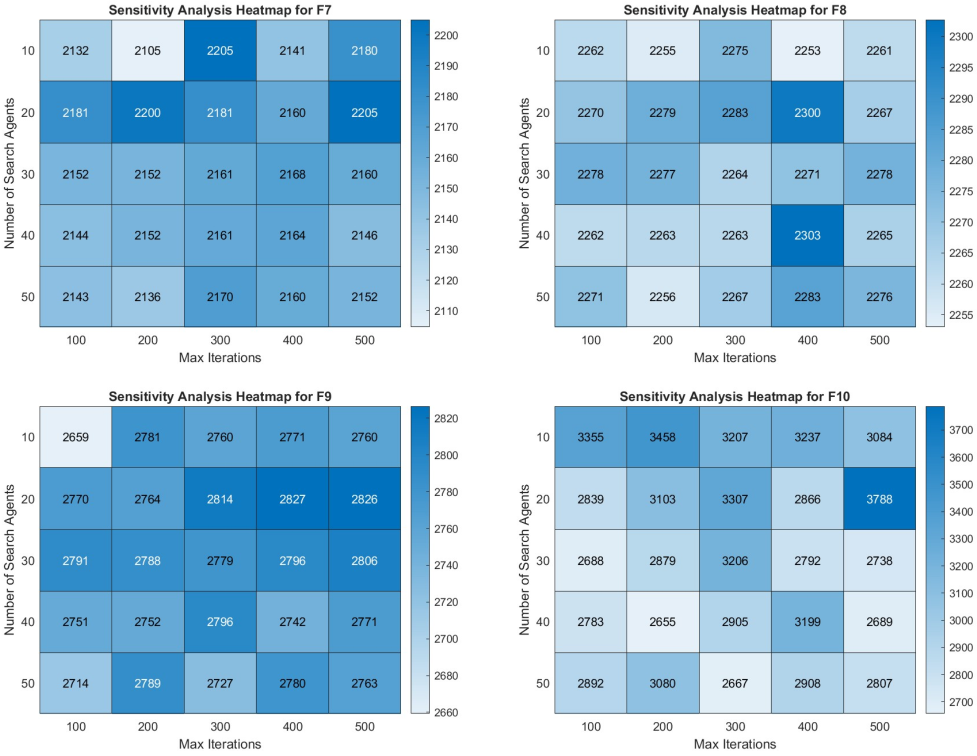

Figure 8. *Cont.*

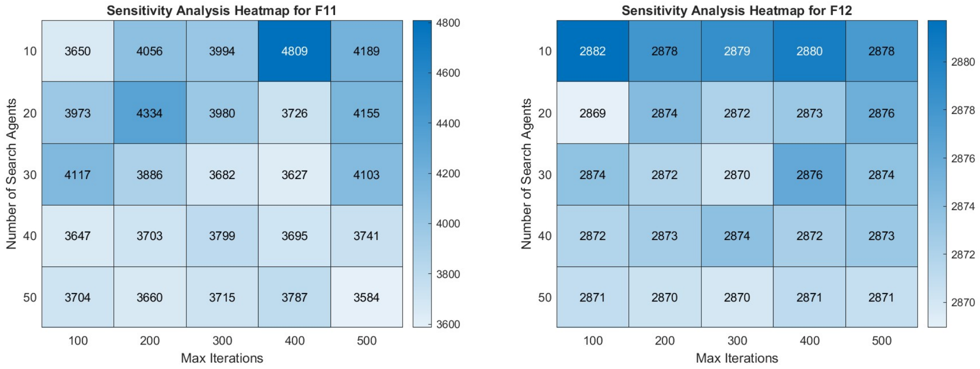

Figure 8. Heatmap analysis over selected functions of CEC2022 (F7–F12).

5.8.4. Box Plot Analysis

The box plot analysis of COASaDE over the CEC2022 benchmark functions (F1–F12) (see Figures 9 and 10) reveals valuable insights into the algorithm's performance variability and robustness. Each box plot presents the distribution of the best fitness scores obtained across multiple runs, showcasing the median, quartiles, and potential outliers. For instance, the box plot for F10 displays a wide range of fitness scores with several outliers, indicating occasional deviations, likely due to the complex landscape of the function. In contrast, functions such as F8 and F9 exhibit more consistent performance with narrower interquartile ranges, suggesting a higher degree of reliability in these cases. The median values in these plots represent the central tendency of the algorithm's performance, while the spread between the quartiles reflects the variability, with the presence of outliers highlighting instances where the algorithm significantly deviates from its typical performance.

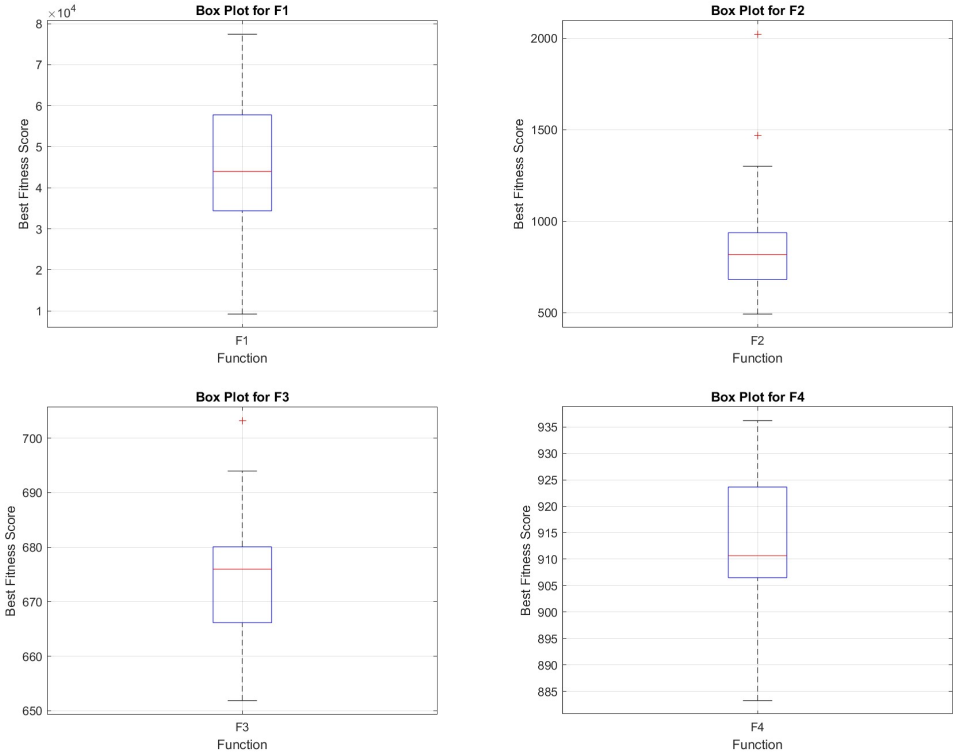

Figure 9. *Cont.*

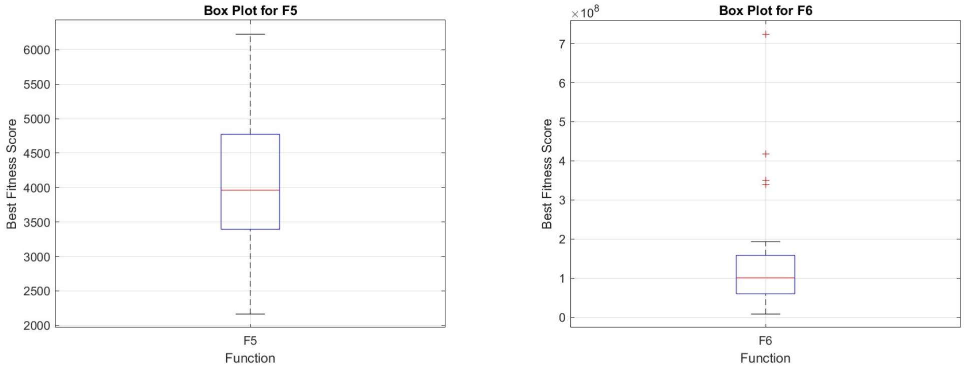

Figure 9. Box plot analysis over selected functions of CEC2022 (F1–F6).

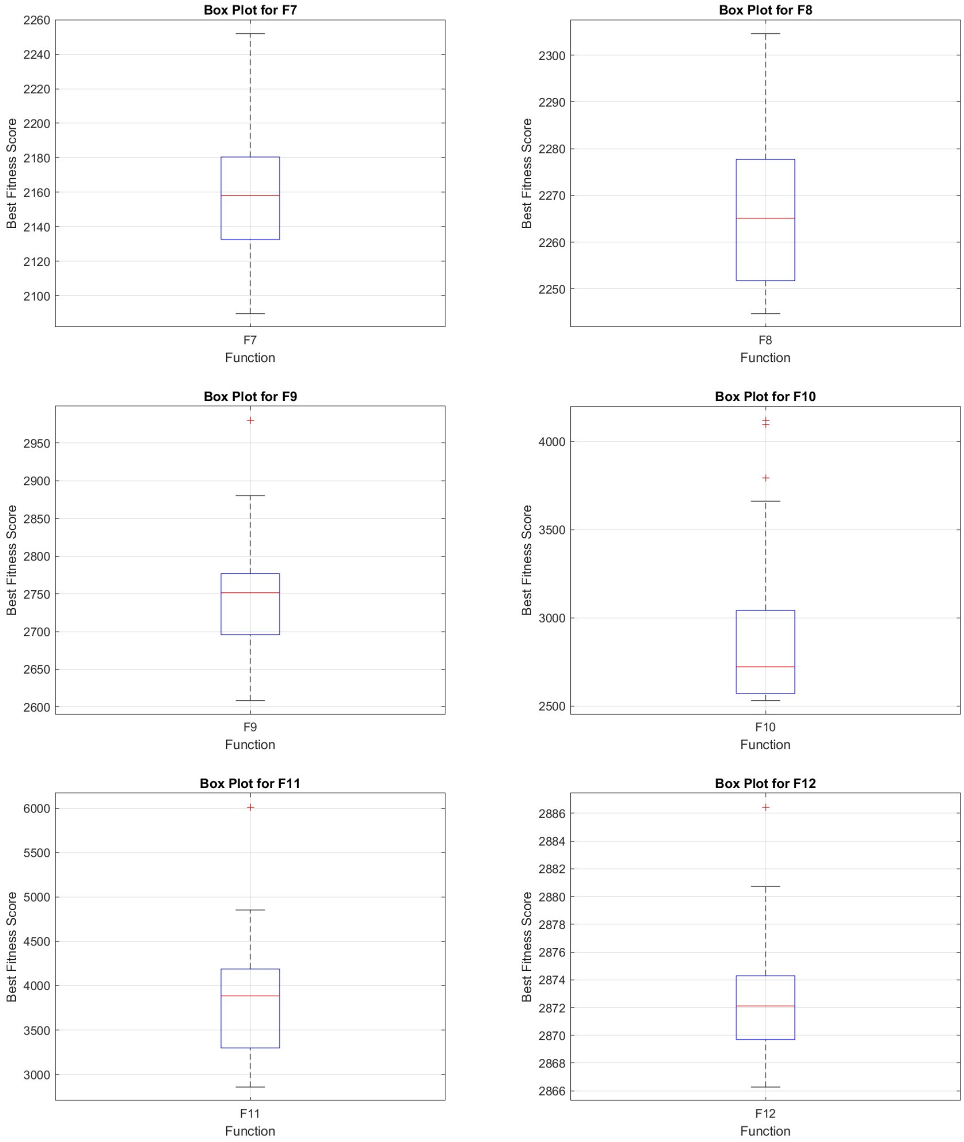

Figure 10. Box plot analysis over selected functions of CEC2022 (F7–F12).

5.9. Histogram Analysis

The histograms provide a detailed view of the distribution of the final fitness values achieved by the COASaDE algorithm across various functions from the CEC2022 benchmark, as shown in Figures 11 and 12. For F1, the histogram shows a spread with a peak around 4 × 10^4, indicating that the algorithm often converges to higher fitness values, with a few instances reaching up to 8 × 10^4. F2 exhibits a broad distribution, with a significant concentration of values between 600 and 1000 and some outliers extending beyond 1200. For F3 the histogram is more centered, with most values clustering around 670–680, indicating relatively consistent performance. The distribution for F4 shows a peak around 900–920, highlighting a narrower range of final fitness values. F5 has a more varied distribution, peaking around 3500–4500, suggesting a wide range of convergence results. For F6, the fitness values are mostly concentrated around lower values, with a significant drop-off as values increase, indicating that the algorithm often finds solutions with lower fitness values. F7 shows a more evenly spread distribution with a central tendency around 2100–2200. For F8 the distribution is highly skewed, with the majority of values around 2250 but with a few instances stretching up to 2400. F9 presents a relatively normal distribution centered around 2700–2800, indicating stable performance. Finally, the histogram for F10 is skewed, with a significant concentration of values around 2400–2800 and a long tail reaching up to 3800, suggesting occasional higher fitness values.

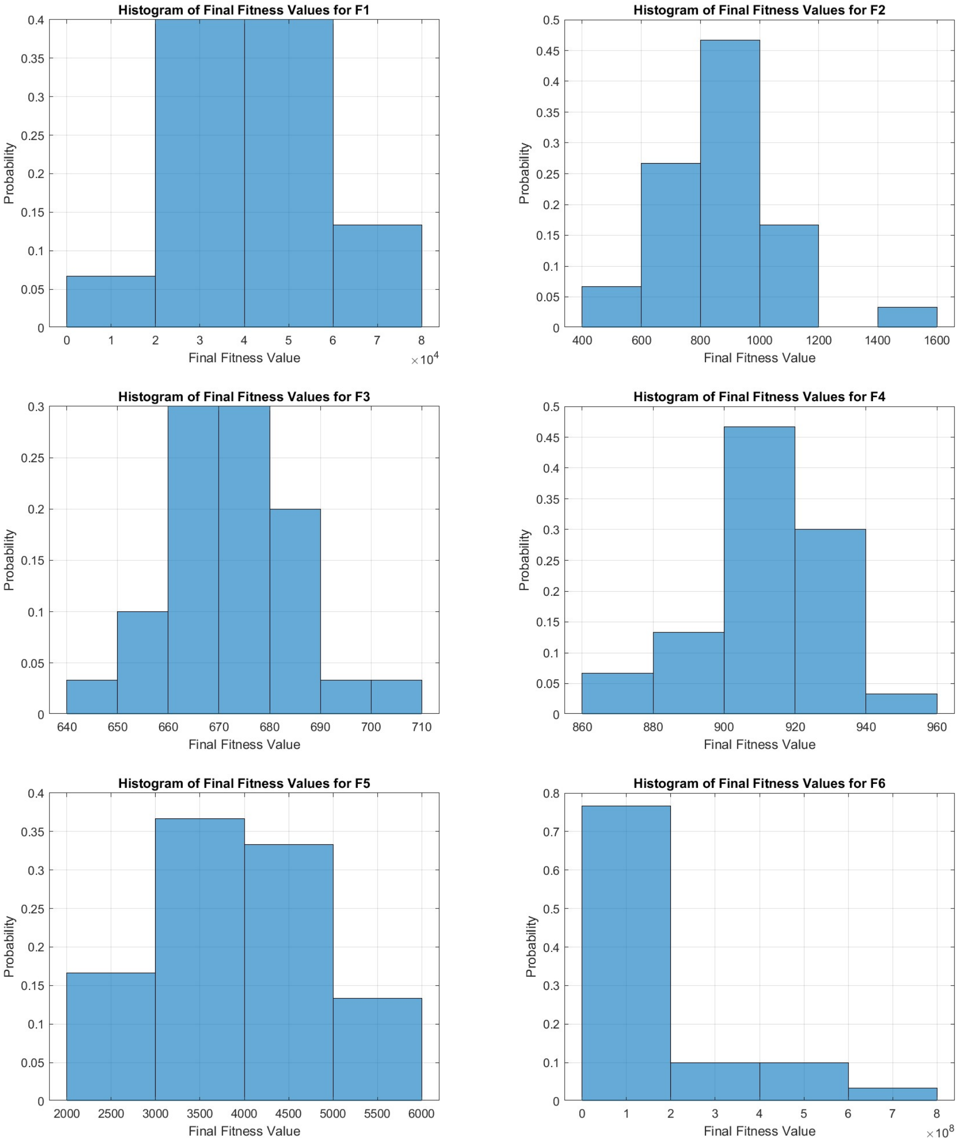

Figure 11. Histogram analysis over selected functions of CEC2022 (F1–F6).

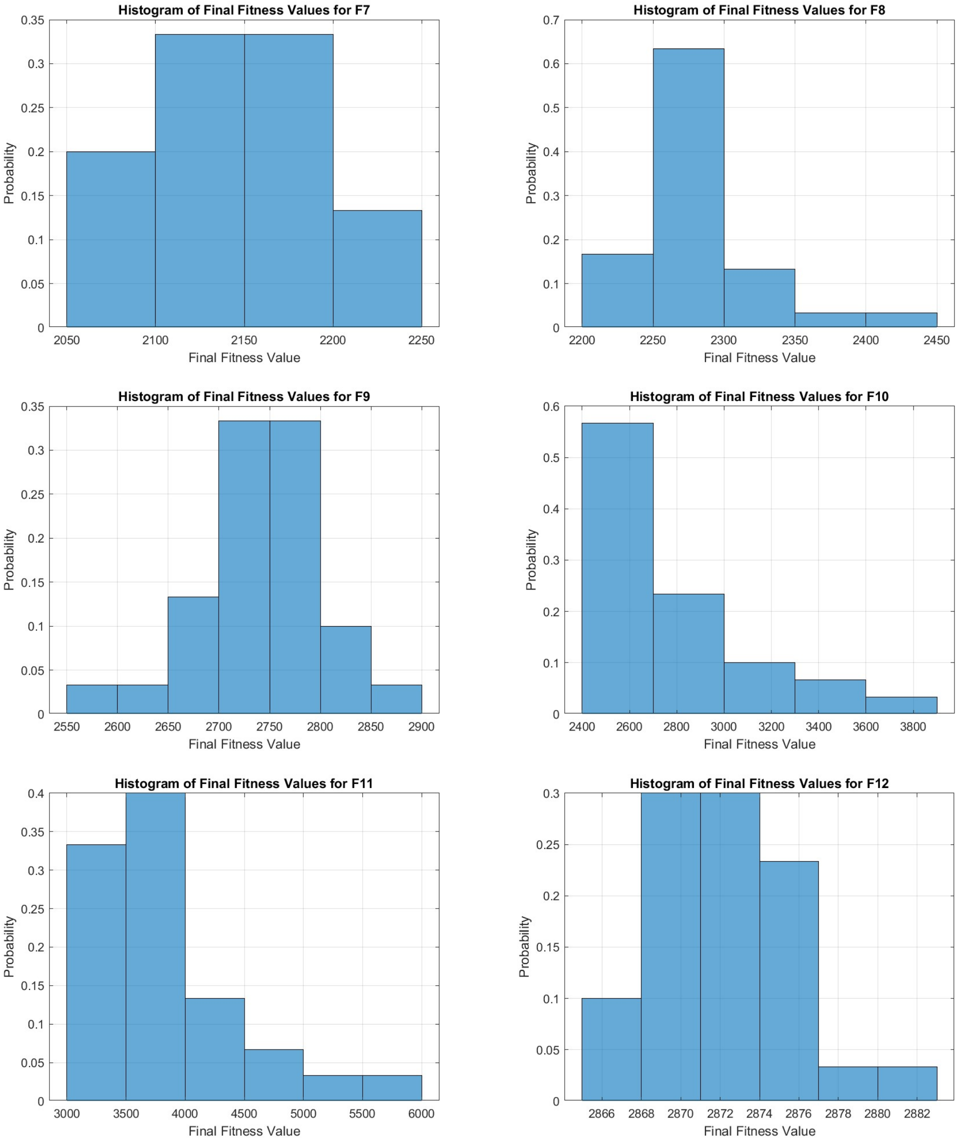

Figure 12. Histogram analysis over selected functions of CEC2022 (F7–F12).

6. Application of COASaD for Solving Engineering Design Problems

6.1. Welded Beam Design Problem

The welded beam design problem represents a complex engineering optimization challenge aimed at identifying optimal beam dimensions that meet specific mechanical constraints while minimizing a cost function. The cost function associated with material and manufacturing expenses is provided by Equation (23):

$$z = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14 + x_2) \tag{23}$$

where x_1, x_2, x_3, and x_4 are the design variables.

The mechanical constraints are formulated based on fundamental principles of mechanical engineering, and are represented by the following Equations (24)–(26).

$$M = P\left(L + \frac{x_2}{2}\right) \tag{24}$$

$$R = \sqrt{\left(\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right)} \tag{25}$$

$$J = 2\sqrt{2}x_1x_2\left(\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right) \tag{26}$$

T shear and bending stresses are calculated using Equations (27)–(29).

$$t_1 = \frac{P}{\sqrt{2}x_1x_2} \tag{27}$$

$$t_2 = \frac{M \cdot R}{J} \tag{28}$$

$$t = \sqrt{t_1^2 + 2t_1t_2\left(\frac{x_2}{2R}\right) + t_2^2} \tag{29}$$

$$s = \frac{6PL}{x_4x_3^2} \tag{30}$$

The deflection and buckling load of the beam are detailed in Equations (31) and (32).

$$d = \frac{4PL^3}{Ex_4x_3^3} \tag{31}$$

$$P_c = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{32}$$

The constraint functions are defined in Equation (33).

$$c = \begin{cases} t - t_{\max} \\ s - s_{\max} \\ x_1 - x_4 \\ 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \\ 0.125 - x_1 \\ d - d_{\max} \\ P - P_c \end{cases} \tag{33}$$

To ensure adherence to these constraints during the optimization process, the penalty functions shown in Equation (34) are employed:

$$f = z + 10^5 \cdot (\mathrm{sum}(v) + \mathrm{sum}(g)) \tag{34}$$

where v is a vector that flags each constraint's violation, ensuring that the design meets mechanical requirements while optimizing cost.

Figure 13 illustrates a welded beam subjected to a point load P. It details dimensions such as the total length L, beam heights h and widths $b = x(1), x(4)$, and weld lengths $l = x(2)$ and thickness values $t = x(3)$. These parameters are critical for assessing the beam's structural integrity and performance under load.

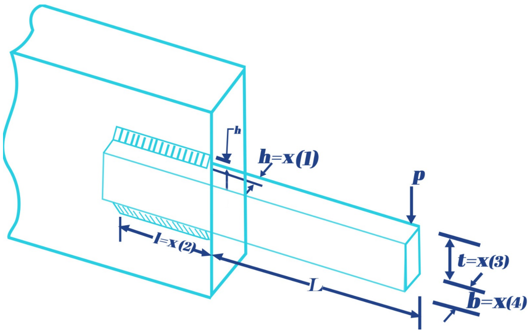

Figure 13. Diagram of the welded beam.

The results achieved by COASaDE for the welded beam design problem (Table 13) demonstrate its superiority compared to several other optimization algorithms. COASaDE achieved the best objective function value (optimal) of 1.674E+00, outperforming all other optimizers in the study. It also maintained the lowest mean objective value (1.674E+00) and standard deviation (8.940E-04), indicating both accuracy and consistency in finding optimal solutions. When comparing COASaDE to other optimizers, notable differences can be observed. While COA had an optimal value of 1.670E+00, very close to that of COASaDE, its mean objective value was higher at 1.731E+00, with a standard deviation of 6.635E-02. This indicates that while COA occasionally found good solutions, it was less consistent than COASaDE. GWO had an optimal of 1.673E+00, also close to COASaDE, but with a higher mean of 1.733E+00 and a larger standard deviation of 1.089E-01, showing more variability in finding optimal solutions compared to COASaDE. AVOA had an optimal of 1.698E+00, with a mean of 1.792E+00 and a standard deviation of 1.422E-01, highlighting AVOA's inconsistency. CSA achieved an optimal of 1.777E+00, with a mean of 1.793E+00 and a standard deviation of 1.366E-02, showing consistency but inferior overall performance. COOT had an optimal of 1.782E+00 with a mean of 1.838E+00 and a standard deviation of 4.840E-02, showing less accuracy and consistency. In terms of computational time, COASaDE performed efficiently with a recorded time of 0.379633, which was competitive with other algorithms such as GWO (0.375307) and AVOA (0.389991) and faster than others such as CSA (0.5748) and DBO (0.902343).

Table 13. Welded beam design problem results.

Optimizer	Optimal	x1	x2	x3	x4	Mean	Std	Min	Max	Time	Rank
COASaDE	1.674E+00	1.982E-01	3.355E+00	9.198E+00	1.990E-01	1.674E+00	8.940E-04	1.674E+00	1.676E+00	0.379633	1
COA	1.670E+00	1.988E-01	3.338E+00	9.192E+00	1.989E-01	1.731E+00	6.635E-02	1.670E+00	1.833E+00	0.416728	2
GWO	1.673E+00	1.995E-01	3.329E+00	9.178E+00	1.995E-01	1.733E+00	1.089E-01	1.673E+00	1.926E+00	0.375307	3
AVOA	1.698E+00	1.766E-01	3.813E+00	9.192E+00	1.988E-01	1.792E+00	1.422E-01	1.698E+00	2.042E+00	0.389991	4
CSA	1.777E+00	1.650E-01	4.231E+00	9.400E+00	2.001E-01	1.793E+00	1.366E-02	1.777E+00	1.809E+00	0.5748	5
COOT	1.782E+00	1.926E-01	3.328E+00	1.000E+01	1.973E-01	1.838E+00	4.840E-02	1.782E+00	1.916E+00	0.377452	6
ChOA	1.784E+00	1.747E-01	3.657E+00	1.000E+01	1.955E-01	1.863E+00	5.167E-02	1.784E+00	1.922E+00	0.446606	7
SCA	1.679E+00	1.910E-01	3.491E+00	9.192E+00	1.989E-01	1.873E+00	2.438E-01	1.679E+00	2.265E+00	0.406579	8
DBO	1.721E+00	2.053E-01	3.220E+00	9.189E+00	2.064E-01	1.976E+00	1.736E-01	1.721E+00	2.194E+00	0.902343	9
HGS	1.732E+00	1.953E-01	3.485E+00	8.913E+00	2.115E-01	1.979E+00	2.393E-01	1.732E+00	2.336E+00	0.708962	10
HHO	1.787E+00	1.806E-01	3.662E+00	9.743E+00	2.000E-01	1.998E+00	1.733E-01	1.787E+00	2.222E+00	0.795756	11
SA	1.801E+00	1.413E-01	4.878E+00	9.188E+00	2.029E-01	1.998E+00	2.133E-01	1.801E+00	2.354E+00	0.901079	12
AO	1.959E+00	6.835E-01	4.449E-01	1.758E+00	5.167E-01	2.253E+00	2.606E-01	1.959E+00	2.542E+00	2.6031	13
HHO	2.131E+00	1.000E-01	7.425E+00	8.509E+00	2.336E-01	2.267E+00	1.300E-01	2.131E+00	2.475E+00	0.467793	14
BO	1.913E+00	1.028E-01	6.841E+00	9.192E+00	1.988E-01	2.273E+00	3.253E-01	1.913E+00	2.587E+00	0.430476	15
BWO	2.147E+00	1.000E-01	9.238E+00	9.187E+00	1.991E-01	2.279E+00	8.143E-02	2.147E+00	2.354E+00	0.963234	16
FOX	2.116E+00	−2.334E+26	-5.457E+23	7.235E+20	5.518E+21	2.295E+00	1.446E-01	2.116E+00	2.423E+00	0.712557	17
WOA	1.964E+00	1.084E-01	6.835E+00	8.982E+00	2.083E-01	2.509E+00	7.561E-01	1.964E+00	3.655E+00	0.373818	18
GA	4.841E+00	1.000E-01	1.000E-01	1.000E-01	1.000E-01	6.198E+00	1.266E+00	4.841E+00	7.620E+00	0.399814	19

6.2. *Pressure Vessel Design Problem*

The pressure vessel design problem is an optimization challenge that involves determining the optimal dimensions of a cylindrical vessel with hemispherical ends. The main objective is to minimize the material cost while ensuring that the structure meets specific structural and dimensional constraints. This task is pivotal in industries where pressure vessels are essential components, such as in chemical processing, power generation, and aerospace applications.

Objective function: The objective function is based on the physical and geometrical properties of the vessel. The function is structured to calculate the total material cost by considering the dimensions and material thickness of the vessel. The cost function is articulated in Equation (36):

$$\begin{aligned} z =& 0.6224 \times 0.0625 \times x_1 \times x_3 \times x_4 + 1.7781 \times 0.0625 \times x_2 \times x_3^2 \\ &+ 3.1661 \times (0.0625 \times x_1)^2 \times x_4 + 19.84 \times (0.0625 \times x_1)^2 \times x_3 \end{aligned} \tag{35}$$

The design variables for the vessel are defined with specific measurements in inches. The variable x_1 represents the thickness of the shell, x_2 corresponds to the thickness of the heads, x_3 is the inner radius of the vessel, and x_4 denotes the length of the cylindrical section of the vessel. Each of these variables is critical for specifying the structural dimensions and ensuring the integrity of the vessel's design.

These variables (x_1, x_2, x_3, and x_4) are crucial as they directly impact both the structural integrity and cost-efficiency of the vessel. The objective of the optimization problem is to determine the values of these variables that minimize the material cost while ensuring adherence to safety and performance standards.

The pressure vessel design problem focuses on determining the optimal dimensions of a cylindrical vessel with hemispherical ends to minimize the material cost while ensuring compliance with structural and dimensional constraints. The design challenge is depicted in Figure 14, which illustrates the vessel and its components.

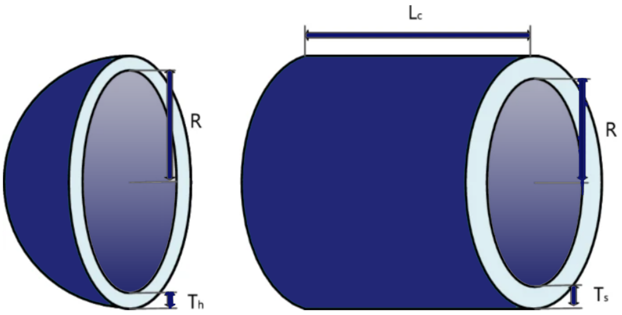

Figure 14. Diagram of cylindrical vessel with hemispherical ends.

Objective Function:

The objective function aims to minimize the cost associated with the materials used in the construction of the pressure vessel. This objective is quantified in the cost function expressed in Equation (36):

$$\begin{aligned} z =& 0.6224 \times 0.0625 \times x_1 \times x_3 \times x_4 + 1.7781 \times 0.0625 \times x_2 \times x_3^2 \\ &+ 3.1661 \times (0.0625 \times x_1)^2 \times x_4 + 19.84 \times (0.0625 \times x_1)^2 \times x_3 \end{aligned} \tag{36}$$

where x_1, x_2, x_3, and x_4 represent the thickness of the shell, the thickness of the heads, the inner radius, and the length of the cylindrical section, respectively.

Constraints:

The constraints are designed to ensure that the vessel's dimensions are both feasible and meet all design requirements, and are detailed in Equations (37)–(44).

$$c_1 = -0.0625 \times x_1 + 0.0193 \times x_3 \tag{37}$$

$$c_2 = -0.0625 \times x_2 + 0.00954 \times x_3 \tag{38}$$

$$c_3 = -\pi \times x_3^2 \times x_4 - \frac{4}{3} \times \pi \times x_3^3 + 1296000 \tag{39}$$

$$c_4 = x_4 - 240 \tag{40}$$

$$c_5 = 1 - x_1 \tag{41}$$

$$c_6 = 1 - x_2 \tag{42}$$

$$c_7 = 10 - x_3 \tag{43}$$

$$c_8 = 10 - x_4 \tag{44}$$

Constraints c_1 and c_2 relate the shell and head thickness to the radius, ensuring structural integrity. Constraint c_3 checks the internal volume against a prescribed limit. Constraints c_4 through c_8 are simple bounding constraints that restrict the size and dimensions of the vessel components.

Penalty function:

The optimization problem utilizes a penalty method in which each constraint violation contributes significantly to the objective function, effectively penalizing infeasible solutions, as shown in Equation (45):

$$f = z + 10^5 \cdot (\text{sum}(v) + \text{sum}(g)) \tag{45}$$

where v is an indicator vector in which each element is set to 1 if the corresponding constraint is violated, helping to steer the optimization process towards feasible designs.

COASaDE achieves an optimal cost of 5.885×10^3, matched by several other high-performance algorithms such as SHIO, SCA, MGO, HOA, HLOA, FVIM, and EAO, demonstrating competitive results. Notably, COASaDE consistently reaches identical optimal design parameters ($x1 = 12.45$, $x2 = 6.154$, $x3 = 40.32$, $x4 = 200$) to those reported by the aforementioned optimizers, indicating high precision in finding the global optimum. In contrast, algorithms such as SSA, POA, and COA exhibit higher optimal costs and significantly varied design parameters, reflecting less efficiency or stability in achieving designs with minimal cost. The consistent values across multiple dimensions by COASaDE and select others highlight the robustness and reliability of these algorithms in converging to an optimal solution, marking them as preferable choices for this specific engineering optimization task.

The results of COASaDE on the pressure vessel design problem (Table 14) highlight its effectiveness compared to other optimization algorithms. COASaDE achieved the optimal objective function value (optimal) of 5.885E+03, outperforming other optimizers with a mean of 6.091E+03 and a standard deviation of 4.492E+02. This indicates both high accuracy and consistency. In comparison, HHO also reached an optimal value of 5.885E+03, but with a higher mean (5.921E+03) and a smaller standard deviation (6.968E+01), showing good consistency but slightly lower overall performance. COA matched the optimal value of 5.885E+03 with an exceptionally low standard deviation (3.008E-02), highlighting remarkable consistency but a slightly lower mean (5.885E+03) than COASaDE. AO had an optimal value of 5.886E+03, with a mean of 5.902E+03 and a standard deviation of 1.841E+01, indicating good but less consistent performance. CSA achieved an optimal value of 5.889E+03, with a higher mean of 6.314E+03 and a standard deviation of 4.464E+02, showing less consistency and accuracy. AVOA obtained an optimal value of 5.901E+03, with a mean of 5.943E+03 and a standard deviation of 4.891E+01, demonstrating more variability. Other algorithms such as SA, ChOA, COOT, DBO, and SCA showed progressively higher optimal values and greater variability, with higher means and standard deviations, indicating less consistent and accurate performance.

Table 14. Pressure vessel design problem results.

Optimizer	Optimal	x1	x2	x3	x4	Min	Mean	Max	Std	Time	Rank
COASaDE	5.885E+03	1.245E+01	6.154E+00	4.032E+01	2.000E+02	5.885E+03	6.091E+03	7.319E+03	4.492E+02	1.312386	1
WOA	5.885E+03	1.245E+01	6.154E+00	4.032E+01	2.000E+02	5.885E+03	5.921E+03	6.085E+03	6.968E+01	1.158006	2
COA	5.885E+03	1.245E+01	6.154E+00	4.032E+01	2.000E+02	5.885E+03	5.885E+03	5.885E+03	3.008E-02	1.628219	3
AO	5.886E+03	1.245E+01	6.156E+00	4.032E+01	1.999E+02	5.886E+03	5.902E+03	5.936E+03	1.841E+01	0.623258	4
CSA	5.889E+03	1.248E+01	6.170E+00	4.042E+01	1.986E+02	5.889E+03	6.314E+03	7.197E+03	4.464E+02	1.200329	5
AVOA	5.901E+03	1.247E+01	6.188E+00	4.037E+01	1.995E+02	5.901E+03	5.943E+03	6.049E+03	4.891E+01	0.58723	6
SA	5.903E+03	1.260E+01	6.233E+00	4.079E+01	1.936E+02	5.903E+03	6.308E+03	6.779E+03	3.346E+02	0.814459	7
ChOA	5.920E+03	1.277E+01	6.310E+00	4.134E+01	1.863E+02	5.920E+03	6.316E+03	7.130E+03	4.471E+02	0.543118	8
COOT	5.947E+03	1.273E+01	6.331E+00	4.118E+01	1.891E+02	5.947E+03	6.419E+03	7.363E+03	5.168E+02	0.730856	9
DBO	5.969E+03	1.316E+01	6.501E+00	4.259E+01	1.706E+02	5.969E+03	7.367E+03	1.572E+04	2.981E+03	1.031498	10
SCA	5.976E+03	1.325E+01	6.549E+00	4.290E+01	1.669E+02	5.976E+03	6.602E+03	7.205E+03	3.892E+02	3.228425	11
HGS	6.137E+03	1.424E+01	7.162E+00	4.611E+01	1.326E+02	6.137E+03	6.612E+03	7.212E+03	3.840E+02	0.707306	12
GWO	6.417E+03	1.269E+01	7.801E+00	4.100E+01	2.000E+02	6.417E+03	7.291E+03	8.326E+03	6.150E+02	0.497152	13
HHO	7.614E+03	1.465E+01	7.496E+00	4.264E+01	2.000E+02	7.614E+03	1.282E+04	1.754E+04	3.202E+03	0.676377	14

6.3. *Spring Design Problem*

The spring design problem is a prevalent optimization challenge in mechanical engineering. It involves determining the optimal dimensions of a coiled spring to minimize the material used in its construction while complying with specific mechanical and design constraints. The problem is typically expressed through the MATLAB function `P3(x)`, which evaluates the performance and feasibility of different spring designs based on their dimensions.

Objective function:

The objective function is formulated to capture the total energy stored in the spring, which is directly affected by the physical dimensions of the spring. This function is strategically designed to minimize the material usage, potentially lowering production costs and enhancing material efficiency. The objective function is defined as Equation (46).

$$z = (x_3 + 2) \times x_2 \times x_1^2 \tag{46}$$

In this formulation, x_1, x_2, and x_3 represent the wire diameter, mean coil diameter, and number of active coils, respectively. These variables are crucial, as they significantly impact the mechanical properties of the spring, such as its stiffness and load-bearing capacity.

Constraints:

The design of the spring is subject to several constraints that are essential for ensuring both functionality and durability. These include stress constraints to prevent material failure, space constraints to ensure the spring fits within a predetermined design envelope, and resonance constraints to avoid vibrational issues that could arise in the operational environment. Each of these factors plays a critical role in the holistic design and successful application of the spring.

These constraints are typically represented as functions of the design variables x_1, x_2, and x_3; they ensure that the spring can perform its intended function safely and effectively. For instance, the stress within the spring must not exceed the yield strength of the material used.

Therefore, the optimization problem involves finding values for x_1, x_2, and x_3 that minimize the objective function while satisfying all imposed constraints. This problem is often solved using numerical methods that explore a range of potential designs to find the most efficient and practical solution.

The optimization of spring design involves determining the optimal dimensions of a coiled spring to ensure minimal material usage while adhering to the strict mechanical and design constraints.

Objective function:

The objective function is designed to minimize the material cost and potential energy stored during compression. This function is expressed in Equation (47):

$$z = (x_3 + 2) \times x_2 \times x_1^2 \tag{47}$$

where x_1, x_2, and x_3 are variables representing the wire diameter, mean coil diameter, and number of active coils, respectively. Each variable significantly contributes to the spring's physical characteristics.

Constraints:

The design constraints atr crucial for maintaining the spring's structural integrity and functionality, and are detailed in Equations (48) through (51).

$$c_1 = \frac{-x_2^3 \times x_3}{71785 \times x_1^4} + 1 \geq 0 \tag{48}$$

$$c_2 = \left(\frac{4x_2^2 - x_1 \times x_2}{12566 \times (x_2 \times x_1^3 - x_1^4)}\right) + \frac{1}{5108 \times x_1^2} - 1 \geq 0 \tag{49}$$

$$c_3 = 1 - \frac{140.45 \times x_1}{x_2^2 \times x_3} \geq 0 \tag{50}$$

$$c_4 = \frac{x_1 + x_2}{1.5} - 1 \geq 0 \tag{51}$$

These constraints ensure that the spring does not succumb to stresses, excessive deflections, or operational limitations. Specifically, constraints c_1 and c_2 address stress and deflection limits critical to the spring's mechanical performance. Constraint c_3 sets a lower bound on the fundamental frequency, ensuring that the spring operates safely within its dynamic range. Additionally, constraint c_4 imposes a dimensional constraint, likely related to the installation space requirements, ensuring that the spring fits appropriately within its designated environment.

Penalty function:

To enforce these constraints effectively within the optimization process, a penalty function is used, as shown in Equation (52):

$$f = z + 10^5 \cdot (\text{sum}(v) + \text{sum}(g)) \tag{52}$$

where v is an indicator vector in which each element is set to 1 if the corresponding constraint is violated, ensuring that noncompliant designs are discouraged and steering the optimization towards feasible and efficient solutions.

The results of COASaDE on the spring design problem (Table 15) highlight its superior performance compared to other optimization algorithms. COASaDE achieved the optimal objective function value (optimal) of 1.267E-02, consistently maintaining this value with a mean and standard deviation both at 1.267E-02 and 3.570E-08, respectively, indicating exceptional accuracy and consistency. In comparison, COA achieved the same optimal value of 1.267E-02 but with a slightly higher mean of 1.268E-02 and a standard deviation of 1.655E-05, showing slightly less consistency. SA matched the optimal value of 1.267E-02 with a mean of 1.269E-02 and a standard deviation of 4.403E-05, indicating good but less consistent performance. GWO also achieved the optimal value of 1.267E-02, with a mean of 1.269E-02 and a standard deviation of 4.496E-05, showing similar variability. CSA matched the optimal value of 1.267E-02 but with a higher mean of 1.314E-02 and a larger standard deviation of 6.621E-04, demonstrating less consistency and accuracy. COOT achieved the optimal value but had an anomalously high mean and standard deviation, indicating instability. Other algorithms, such as ChOA, WOA, HGS, AO, AVOA, DBO, SCA, and HHO, showed progressively higher optimal values and greater variability, indicating less consistent and accurate performance. The ranking in the table underscores COASaDE's top performance, securing the first rank and highlighting its robustness, precision, and efficiency in solving the spring design optimization problem.

Table 15. Spring design problem results.

Optimizer	Optimal	Variable 1	Variable 2	Variable 3	Min	Mean	Max	Std	Time	Rank
COASaDE	1.267E-02	5.169E-02	3.567E-01	1.129E+01	1.267E-02	1.267E-02	1.267E-02	3.570E-08	0.944562	1
COA	1.267E-02	5.164E-02	3.556E-01	1.135E+01	1.267E-02	1.268E-02	1.272E-02	1.655E-05	0.827274	2
SA	1.267E-02	5.144E-02	3.508E-01	1.165E+01	1.267E-02	1.269E-02	1.281E-02	4.403E-05	1.246854	3
GWO	1.267E-02	5.187E-02	3.612E-01	1.103E+01	1.267E-02	1.269E-02	1.281E-02	4.496E-05	0.724503	4
CSA	1.267E-02	5.198E-02	3.637E-01	1.089E+01	1.267E-02	1.314E-02	1.431E-02	6.621E-04	1.592948	5
COOT	1.267E-02	5.204E-02	3.651E-01	1.081E+01	1.267E-02	1.517E+05	1.517E+06	4.798E+05	0.81524	6
ChOA	1.267E-02	5.101E-02	3.406E-01	1.230E+01	1.267E-02	1.313E-02	1.554E-02	9.834E-04	2.219108	7
WOA	1.267E-02	5.100E-02	3.404E-01	1.232E+01	1.267E-02	1.303E-02	1.409E-02	5.279E-04	0.699102	8
HGS	1.267E-02	5.108E-02	3.422E-01	1.219E+01	1.267E-02	1.352E-02	1.520E-02	7.506E-04	1.054886	9
AO	1.268E-02	5.100E-02	3.404E-01	1.232E+01	1.268E-02	1.272E-02	1.293E-02	7.665E-05	0.504661	10
AVOA	1.269E-02	5.100E-02	3.402E-01	1.234E+01	1.269E-02	1.274E-02	1.298E-02	8.709E-05	0.745025	11
DBO	1.275E-02	5.100E-02	3.391E-01	1.245E+01	1.275E-02	1.710E-02	1.857E-02	2.246E-03	0.668569	12
SCA	1.282E-02	5.205E-02	3.651E-01	1.096E+01	1.282E-02	1.313E-02	1.346E-02	1.875E-04	0.672099	13
HHO	1.394E-02	5.100E-02	3.268E-01	1.440E+01	1.394E-02	1.575E-02	3.051E-02	5.186E-03	0.644913	14

6.4. Speed Reducer Design Problem

The speed reducer design problem is a complex optimization task centered around minimizing the cost function associated with the design of a gear train within a speed reducer. This problem encapsulates a myriad of mechanical design constraints, aiming to achieve an optimal balance between material cost, geometry, and structural limits. The objective is to design a speed reducer that is cost-effective, structurally sound, and operationally efficient.

Objective function:

The objective function for the speed reducer design problem incorporates various geometric and material cost factors, reflecting the complex interdependencies of the gear train components. The function is formulated as shown in Equation (53) .

$$\begin{aligned} z =& 0.7854x_1(x_2^2)(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) \\ &+ 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \end{aligned} \tag{53}$$

This equation captures the essential cost elements and mechanical requirements integral to designing an efficient speed reducer.

This function is meticulously crafted to account for various factors. Primarily, it addresses the costs associated with the material used in the gear train, which is significantly influenced by dimensions such as face width (x_1), gear tooth thickness (x_2), and gear diameters (x_3, x_4, x_5). Additionally, the function considers the structural integrity and operational efficacy, which are modulated by variables x_6 and x_7. These variables represent the gear diameters, and are crucial for ensuring the gear's mechanical performance.

The careful selection of these terms ensures that the function reflects the trade-offs between material strength, weight, and overall cost of the speed reducer. The objective needs to not just minimize the cost but also ensure that the speed reducer meets all operational and safety standards.

Constraints:

The design of the speed reducer must also adhere to a series of constraints which ensure that the product is not only economically viable but also meets all required performance criteria. These constraints include but are not limited to stress and strain limits, dimensional tolerances, and compatibility requirements for interfacing with other mechanical systems. Each constraint is typically formulated as a function of the design variables, and is critical in guiding the optimization process towards feasible solutions.

The speed reducer design problem is a critical engineering optimization challenge aimed at minimizing a cost function associated with the gear train design within a speed reducer. The objective is to balance material costs, geometric considerations, and structural limits. Figure 15 illustrates the complex interplay of these factors.

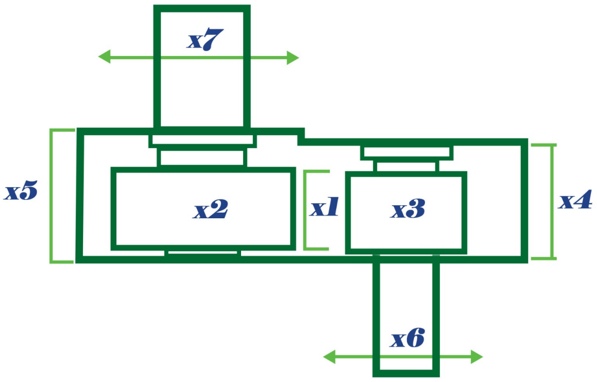

Figure 15. Design of the speed reducer gear train.

Objective function:

The objective function integrates the material costs and geometric dependencies of various components, as expressed in Equation (54).

$$\begin{aligned} z =& 0.7854x_1(x_2^2)(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) \\ &+ 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \end{aligned} \quad (54)$$

This function is designed to minimize the material cost while considering the weight and durability of the speed reducer's components.

Constraints:

The design constraints, which are essential for ensuring the design's feasibility and efficiency, are formulated as shown below and illustrated in Figure 15.

$$c_1 = \frac{27}{x_1x_2^2x_3} - 1 \quad (55)$$

$$c_2 = \frac{397.5}{x_1x_2^2x_3^2} - 1 \quad (56)$$

$$c_3 = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \quad (57)$$

$$c_4 = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \quad (58)$$

$$c_5 = \frac{1}{110x_6^3}\left(\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6\right)^{0.5} - 1 \quad (59)$$

$$c_6 = \frac{1}{85x_7^3}\left(\left(\frac{745x_5}{x_2x_3}\right)^2 + 157.5 \times 10^6\right)^{0.5} - 1 \quad (60)$$

$$c_7 = \frac{x_2x_3}{40} - 1 \quad (61)$$

$$c_8 = \frac{5x_2}{x_1} - 1 \quad (62)$$

$$c_9 = \frac{x_1}{12x_2} - 1 \quad (63)$$

$$c_{10} = \frac{1.5x_6 + 1.9}{x_4} - 1 \quad (64)$$

$$c_{11} = \frac{1.1x_7 + 1.9}{x_5} - 1 \quad (65)$$

These constraints ensure that the speed reducer can withstand bending stresses, maintain surface durability, resist stress concentration, and fit geometrically within the overall design.

Penalty function

To ensure compliance with these constraints, a penalty function is employed, as shown in Equation (66):

$$f = z + 10^5 \cdot (\text{sum}(v) + \text{sum}(g)) \tag{66}$$

where v is an indicator vector that flags each constraint violation to ensure that designs not meeting the safety and performance standards are penalized, thereby guiding the optimization process towards feasible solutions.

Table 16 shows that COASaDE achieved the best score (Best Score) of 2.994E+03, maintaining a low mean score of 2.998E+03 and a standard deviation of 1.242E+01 on the speed reducer design problem. In comparison, COA achieved the same best score of 2.994E+03 with an exceptionally low standard deviation of 1.104E-12, demonstrating remarkable consistency, although its recorded time was significantly higher at 2.472202. ChOA matched the best score of 2.994E+03 but with a higher mean of 3.000E+03 and a standard deviation of 9.610E+00, showing slightly more variability. CSA also achieved the best score of 2.994E+03 with a minimal standard deviation of 1.696E-08 and a slightly higher mean, indicating very consistent performance. AO matched the best score but had a higher mean and standard deviation, reflecting less precision. Other algorithms such as WOA, SA, AVOA, HGS, DBO, COOT, GWO, and HHO showed progressively higher best scores and greater variability, indicating less consistent and accurate performance.

Table 16. Speed reducer design problem results.

Optimizer	Best Score	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5	Variable 6	Variable 7	Min	Mean	Max	Std	Time	Rank
COASaDE	2.994E+03	3.500E+00	7.000E-01	1.700E+01	7.300E+00	7.715E+00	3.350E+00	5.287E+00	2.994E+03	2.998E+03	3.034E+03	1.242E+01	0.557706	1
COA	2.994E+03	3.500E+00	7.000E-01	1.700E+01	7.300E+00	7.715E+00	3.350E+00	5.287E+00	2.994E+03	2.994E+03	2.994E+03	1.104E-12	2.472202	2
ChOA	2.994E+03	3.500E+00	7.000E-01	1.700E+01	7.300E+00	7.715E+00	3.350E+00	5.287E+00	2.994E+03	3.000E+03	3.017E+03	9.610E+00	1.451612	3
CSA	2.994E+03	3.500E+00	7.000E-01	1.700E+01	7.300E+00	7.715E+00	3.350E+00	5.287E+00	2.994E+03	2.994E+03	2.994E+03	1.696E-08	1.317274	4
AO	2.994E+03	3.500E+00	7.000E-01	1.700E+01	7.300E+00	7.715E+00	3.350E+00	5.287E+00	2.994E+03	2.994E+03	2.994E+03	1.000E-02	0.694945	5
CSA	2.994E+03	3.500E+00	7.000E-01	1.700E+01	7.300E+00	7.716E+00	3.350E+00	5.287E+00	2.994E+03	2.995E+03	2.995E+03	2.458E-02	1.345214	6
WOA	2.995E+03	3.500E+00	7.000E-01	1.700E+01	7.306E+00	7.716E+00	3.350E+00	5.287E+00	2.995E+03	3.003E+03	3.020E+03	8.954E+00	1.373817	7
SA	2.996E+03	3.500E+00	7.000E-01	1.700E+01	7.302E+00	7.731E+00	3.351E+00	5.287E+00	2.996E+03	2.999E+03	3.012E+03	5.676E+00	0.886892	8
AVOA	3.006E+03	3.506E+00	7.000E-01	1.701E+01	7.300E+00	7.950E+00	3.354E+00	5.288E+00	3.006E+03	3.012E+03	3.021E+03	5.014E+00	0.588294	9
HGS	3.012E+03	3.502E+00	7.000E-01	1.700E+01	7.545E+00	8.290E+00	3.357E+00	5.287E+00	3.012E+03	3.035E+03	3.057E+03	1.328E+01	0.736164	10
DBO	3.013E+03	3.500E+00	7.000E-01	1.700E+01	8.224E+00	8.181E+00	3.352E+00	5.287E+00	3.013E+03	3.526E+03	5.509E+03	8.822E+02	0.887727	11
COOT	3.015E+03	3.505E+00	7.000E-01	1.700E+01	7.410E+00	8.000E+00	3.375E+00	5.295E+00	3.015E+03	3.031E+03	3.060E+03	1.410E+01	0.790502	12
GWO	3.087E+03	3.600E+00	7.000E-01	1.700E+01	8.300E+00	7.790E+00	3.367E+00	5.347E+00	3.087E+03	3.154E+03	3.197E+03	3.692E+01	0.527177	13
HHO	3.096E+03	3.600E+00	7.000E-01	1.700E+01	7.300E+00	8.300E+00	3.486E+00	5.308E+00	3.096E+03	3.150E+03	3.206E+03	3.942E+01	0.71506	14

6.5. *Cantiliver Design Problem*

The cantilever design problem is an engineering challenge that involves optimizing the dimensions of a cantilever structure. The goal is to minimize the material cost while ensuring structural stability under various loading conditions. This optimization problem is critical in fields such as civil engineering, aerospace, and mechanical systems, where cantilevers are frequently used components.

Objective function:

The objective function is specifically designed to represent the total material cost associated with the construction of the cantilever. This function is quantitatively formulated as shown in Equation (67).

$$z = 0.0624 \times (x_1 + x_2 + x_3 + x_4 + x_5) \tag{67}$$

In this model, x_1, x_2, x_3, x_4, and x_5 correspond to the dimensions of the cantilever, such as the thickness or length of different segments. Each variable contributes linearly to the material cost, indicating that any changes in these dimensions will have a direct impact on the overall cost of the structure.

Constraints:

To ensure the structural stability of the cantilever under various loads, several critical constraints must be considered. These include stress constraints designed to prevent material failure under maximum expected loads, deflection constraints aimed at ensuring the cantilever does not deform excessively under load, and natural frequency constraints intended to avoid resonant frequencies that could lead to structural failure. Each of these constraints plays a vital role in the design and safety assurance of the cantilever structure.

These constraints are generally formulated as functions of the design variables, ensuring that the cantilever remains both economical and safe under operational conditions.

Therefore, the design problem involves not only minimizing the objective function but also satisfying a series of structural and performance-based constraints. This holistic approach ensures the creation of a cantilever that is both cost effective and robust in its application.

The cantilever design problem (see Figure 16) is a critical challenge in structural engineering, where the goal is to optimize the dimensions of a cantilever to minimize material costs while ensuring that the structure can withstand operational loads without failure. The optimization is particularly focused on balancing cost efficiency with the structural integrity required to handle expected stresses and deflections.

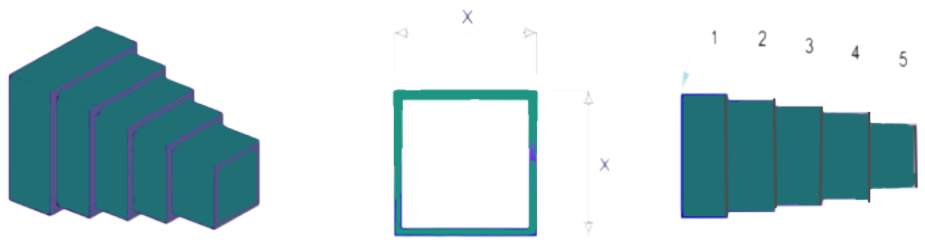

Figure 16. Cantilever design problem.

Objective function:

The objective function is calculated to determine the total cost of materials used in the construction of the cantilever based on the dimensions of its various segments, as shown in Equation (68):

$$z = 0.0624 \times (x_1 + x_2 + x_3 + x_4 + x_5) \tag{68}$$

where x_1, x_2, x_3, x_4, and x_5 might correspond to the thicknesses or lengths of different segments of the cantilever, where each variable affects the material cost linearly.

Constraints:

The cantilever must meet specific structural stiffness and strength criteria, which are formulated in constraint shown in the following Equation (69).

$$c_1 = \left(\frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} \right) - 1 \geq 0 \tag{69}$$

This constraint is derived from the bending stiffness or strength requirements, where the terms reflect the inverse cube relationship between dimension (likely thickness) and bending stiffness or strength for each segment, thereby ensuring the cantilever's adequacy in terms of stiffness and strength.

Penalty function:

To enforce the structural requirements and discourage noncompliance, the penalty function detailed in Equation (70) is utilized.

$$f = z + 10^5 \cdot (\mathrm{sum}(v) + \mathrm{sum}(g)) \tag{70}$$

In this function, v is an indicator vector in which each element is set to 1 if the corresponding constraint is violated. This setup imposes a high cost on designs that do not meet the structural criteria, thereby promoting compliance with the required standards.

As can be seen in Table 17, COASaDE achieved an optimal objective function value (optimal) of 1.340E+00, consistently maintaining this value with a mean of 1.340E+00 and a standard deviation of 2.171E-05, indicating exceptional accuracy and consistency. In comparison, CSA achieved the same optimal value of 1.340E+00 with a slightly higher standard deviation of 1.005E-04, demonstrating good consistency but slightly less precision. GWO also matched the optimal value with a standard deviation of 8.980E-05, showing comparable performance to CSA. COA achieved the optimal value with a mean of 1.340E+00 and a higher standard deviation of 3.502E-04, reflecting more variability. HGS and DBO both matched the optimal value but showed higher means and standard deviations, indicating less consistent performance. Other algorithms such as COOT, HHO, AVOA, AO, SA, ChOA, BWO, SCA, and WOA demonstrated progressively higher optimal values, means, and standard deviations, indicating less consistent and accurate performance.

Table 17. Cantiliver design problem results.

Optimizer	Optimal	x1	x2	x3	x4	Mean	Std	Min	Max	Time	Rank
COASaDE	1.340E+00	6.027E+00	5.300E+00	4.485E+00	3.506E+00	1.340E+00	2.171E-05	1.340E+00	1.340E+00	0.276578	1
CSA	1.340E+00	6.021E+00	5.285E+00	4.504E+00	3.499E+00	1.340E+00	1.005E-04	1.340E+00	1.340E+00	0.239871	2
GWO	1.340E+00	6.027E+00	5.274E+00	4.505E+00	3.499E+00	1.340E+00	8.980E-05	1.340E+00	1.340E+00	0.25402	3
COA	1.340E+00	5.983E+00	5.302E+00	4.502E+00	3.530E+00	1.340E+00	3.502E-04	1.340E+00	1.341E+00	0.282362	4
HGS	1.340E+00	6.055E+00	5.304E+00	4.477E+00	3.492E+00	1.341E+00	6.407E-04	1.340E+00	1.342E+00	0.258944	5
DBO	1.340E+00	6.046E+00	5.269E+00	4.445E+00	3.511E+00	1.341E+00	4.527E-04	1.340E+00	1.342E+00	0.29273	6
COOT	1.341E+00	6.011E+00	5.444E+00	4.437E+00	3.456E+00	1.341E+00	1.117E-03	1.341E+00	1.343E+00	0.259909	7
HHO	1.340E+00	5.997E+00	5.253E+00	4.590E+00	3.450E+00	1.343E+00	2.584E-03	1.340E+00	1.346E+00	0.586414	8
AVOA	1.340E+00	6.060E+00	5.267E+00	4.497E+00	3.441E+00	1.344E+00	3.639E-03	1.340E+00	1.349E+00	0.586687	9
AO	1.343E+00	6.056E+00	5.175E+00	4.722E+00	3.399E+00	1.346E+00	2.847E-03	1.343E+00	1.350E+00	0.529126	10
SA	1.343E+00	5.847E+00	5.150E+00	4.604E+00	3.732E+00	1.352E+00	1.224E-02	1.343E+00	1.374E+00	0.468416	11
ChOA	1.346E+00	5.793E+00	5.425E+00	4.658E+00	3.617E+00	1.355E+00	8.011E-03	1.346E+00	1.366E+00	0.498574	12
BWO	1.344E+00	5.772E+00	5.501E+00	4.719E+00	3.406E+00	1.364E+00	1.372E-02	1.344E+00	1.382E+00	0.325334	13
SCA	1.368E+00	5.778E+00	5.695E+00	4.167E+00	3.735E+00	1.405E+00	2.244E-02	1.368E+00	1.424E+00	0.247594	14
WOA	1.365E+00	7.128E+00	4.988E+00	4.352E+00	3.418E+00	1.419E+00	4.823E-02	1.365E+00	1.471E+00	0.242587	15

6.6. I-Beam Design Problem

The I-beam design problem involves optimizing the geometry of an I-beam to achieve minimal deflection under a specific load while ensuring that the beam's cross-sectional area meets certain size requirements. This problem exemplifies the essential balance between efficient material use and optimal structural performance in beam design.

Objective function:

The objective function is formulated to minimize the beam's deflection by maximizing the section modulus, which is a measure of the beam's resistance to bending. This is mathematically represented by the inverse of the section modulus, as shown in Equation (71):

$$z = \frac{5000}{\left(\frac{tw\cdot(h-2\cdot tf)^3}{12} + \frac{b\cdot tf^3}{6} + 2\cdot b\cdot tf\cdot\left(\left(\frac{h-tf}{2}\right)^2\right)\right)} \tag{71}$$

where b represents the width of the beam, h is the overall height, tw is the web thickness, and tf is the flange thickness. The objective is to minimize z, thereby reducing the deflection under load.

Constraint:

As illustrated in Figure 17, the design includes a constraint to ensure that the total material used for the beam does not fall below a specific threshold, which is important for maintaining structural integrity and compliance with industry standards. The constraint is formulated in the following Equation (72).

$$c_1 = 2\cdot b\cdot tw + tw\cdot(h-2\cdot tf) - 300 \geq 0 \tag{72}$$

This constraint is interpreted as ensuring a minimum cross-sectional area or weight of the beam, which is necessary to meet safety regulations and structural requirements.

Penalty function:

To enforce compliance with this constraint, a penalty function is employed, as detailed in Equation (73):

$$f = z + 10^5 \cdot (\text{sum}(v) + \text{sum}(g)) \tag{73}$$

where v is an indicator vector in which each element is set to 1 if the corresponding constraint is violated. This approach ensures that designs not meeting the required standards incur a significant cost, thereby steering the optimization process towards feasible and compliant designs.

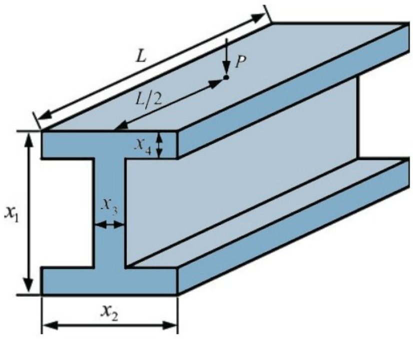

Figure 17. Schematic of the optimized I-beam geometry [41].

As can be seen in Table 18, COASaDE achieved the best score of 6.626E-03, consistently maintaining this value with an incredibly low standard deviation of 1.042E-18, indicating exceptional accuracy and consistency. COA matched this best score with an equally low standard deviation of 9.143E-19, demonstrating remarkable consistency. CSA, ChOA, and SCA also matched the best scores with similarly low standard deviations, reflecting their consistency in performance. DBO achieved the same best score with a slightly higher standard deviation of 6.176E-09, showing minor variability. WOA, AO, AVOA, SA, HGS, and COOT all matched the best scores with low standard deviations, although slightly higher than those of COA and CSA, indicating good consistency but slightly less precision.

Table 18. I-beam design problem results.

Optimizer	Best Score	Variable 1	Variable 2	Variable 3	Variable 4	Min	Mean	Max	Std	Time	Rank
COASaDE	6.626E-03	5.000E+01	8.000E+01	1.765E+00	5.000E+00	6.626E-03	6.626E-03	6.626E-03	1.042E-18	1.060202	1
COA	6.626E-03	5.000E+01	8.000E+01	1.765E+00	5.000E+00	6.626E-03	6.626E-03	6.626E-03	9.143E-19	0.454492	2
CSA	6.626E-03	5.000E+01	8.000E+01	1.765E+00	5.000E+00	6.626E-03	6.626E-03	6.626E-03	9.143E-19	1.025612	3
ChOA	6.626E-03	5.000E+01	8.000E+01	1.765E+00	5.000E+00	6.626E-03	6.626E-03	6.626E-03	9.143E-19	0.984948	4
SCA	6.626E-03	5.000E+01	8.000E+01	1.765E+00	5.000E+00	6.626E-03	6.626E-03	6.626E-03	9.143E-19	2.291116	5
DBO	6.626E-03	5.000E+01	8.000E+01	1.765E+00	5.000E+00	6.626E-03	6.626E-03	6.626E-03	6.176E-09	0.823816	6
WOA	6.626E-03	5.000E+01	8.000E+01	1.765E+00	5.000E+00	6.626E-03	6.626E-03	6.626E-03	9.143E-19	0.543795	7
AO	6.626E-03	5.000E+01	8.000E+01	1.765E+00	5.000E+00	6.626E-03	6.626E-03	6.626E-03	1.075E-11	0.576118	8
AVOA	6.626E-03	5.000E+01	8.000E+01	1.765E+00	5.000E+00	6.626E-03	6.626E-03	6.626E-03	4.446E-08	0.437802	9
SA	6.626E-03	5.000E+01	8.000E+01	1.765E+00	5.000E+00	6.626E-03	6.626E-03	6.626E-03	1.902E-08	0.684637	10
HGS	6.626E-03	5.000E+01	8.000E+01	1.765E+00	5.000E+00	6.626E-03	6.626E-03	6.626E-03	9.691E-08	0.57949	11
COOT	6.626E-03	5.000E+01	8.000E+01	1.765E+00	5.000E+00	6.626E-03	6.626E-03	6.626E-03	4.473E-08	0.620857	12
GWO	6.626E-03	5.000E+01	8.000E+01	1.764E+00	5.000E+00	6.626E-03	6.628E-03	6.632E-03	2.062E-06	0.460884	13
HHO	6.627E-03	5.000E+01	8.000E+01	1.763E+00	5.000E+00	6.627E-03	6.627E-03	6.629E-03	7.931E-07	0.510059	14

Furthermore, COASaDE successfully achieved the optimal value of 6.626×10^{-3}, precisely matching the optimal design variables ($x1 = 50.00$, $x2 = 80.00$, $x3 = 1.765$, $x4 = 5.00$) identified by the vast majority of other optimizers, including WSO, SSA, SHIO, SCA, and EAO, which demonstrates consensus on the solution's accuracy and effectiveness across different algorithms. This uniformity in achieving optimal values and design variables highlights the robustness and reliability of these algorithms in precisely solving structural design optimizations under defined constraints. Interestingly, all algorithms, including MPA, exhibited close results with almost negligible variations, indicating that these optimization techniques are well suited for engineering applications where precision is crucial. Notably, COASaDE stands out with a perfect standard deviation of 1.796×10^{-18}, suggesting its superior ability to consistently reach the exact optimal solution on every run, setting a benchmark in solution consistency among its peers.

6.7. Three-Bar Design Problem

The three-bar truss design problem is a classic engineering optimization challenge that focuses on finding the optimal dimensions of a truss structure. The objective is to minimize the cost of the structure while ensuring that it performs adequately under specified load constraints, embodying a balance between economic efficiency and structural integrity.

Objective function:

The objective function for this problem calculates the cost associated with the truss, where the dimensions directly influence material usage and consequently the overall cost. This relationship is formalized in the following Equation (74):

$$z = (2\sqrt{2}x_1 + x_2) \times 100 \tag{74}$$

where x_1 and x_2 represent the cross-sectional areas of the truss members. This formula incorporates coefficients that likely reflect the cost per unit length of the materials used adjusted for the truss's specific geometric configuration.

Constraints:

The design must meet several constraints in order to ensure that the stress in each member remains within safe limits, as detailed in Equations (75)–(77).

$$c_1 = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} \times 2 - 2 \geq 0 \tag{75}$$

$$c_2 = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} \times 2 - 2 \geq 0 \tag{76}$$

$$c_3 = \frac{1}{x_1 + \sqrt{2}x_2} \times 2 - 2 \geq 0 \tag{77}$$

These constraints are derived from the stress distribution formulas in the truss members, ensuring that the stress does not exceed the material's yield strength. Each constraint addresses different loading conditions and geometric relationships in the truss design, which is crucial for maintaining safety and functionality.

Penalty function:

To ensure compliance with these constraints, the penalty functions shown in Equation (78) is employed.

$$f = z + 10^5 \cdot (\text{sum}(v) + \text{sum}(g)) \tag{78}$$

In this function, v is an indicator vector in which each element is set to 1 if the corresponding constraint is violated. This setup imposes a significant cost on noncompliant designs, effectively steering the optimization algorithm towards solutions that satisfy all performance criteria, thereby ensuring both safety and cost efficiency.

As can be seen in Table 19, COASaDE achieved the minimum value of 2.639E+02, consistently maintaining this value with a mean of 2.639E+02 and an incredibly low standard deviation of 8.994E-12, indicating exceptional accuracy and consistency. CSA matched the optimal value of 2.639E+02 with a slightly higher standard deviation of 1.067E-10, demonstrating very good consistency. GWO also matched the optimal value but had a higher standard deviation of 8.707E-04, indicating more variability. COA achieved the optimal value with a higher standard deviation of 4.776E-03, reflecting more variability. GOA and COOT both matched the optimal value but with even higher standard deviations of 8.400E-03 and 9.651E-03, respectively, showing less precision. Other algorithms such as AVOA, HHO, DBO, ChOA, SCA, BO, SA, BWO, and WOA demonstrated progressively higher optimal values, means, and standard deviations, indicating less consistent and accurate performance. The ranking in the table reinforces COASaDE's top performance, securing the first rank and highlighting its robustness, precision, and efficiency in solving the three-bar design optimization problem.

Table 19. Three-bar design problem results.

Optimizer	Optimal	x1	x2	Mean	Std	Min	Max	Time	Rank
COASaDE	2.639E+02	7.887E-01	4.082E-01	2.639E+02	8.994E-12	2.639E+02	2.639E+02	0.372215	1
CSA	2.639E+02	7.887E-01	4.082E-01	2.639E+02	1.067E-10	2.639E+02	2.639E+02	0.372444	2
GWO	2.639E+02	7.884E-01	4.090E-01	2.639E+02	8.707E-04	2.639E+02	2.639E+02	0.409853	3
COA	2.639E+02	7.880E-01	4.101E-01	2.639E+02	4.776E-03	2.639E+02	2.639E+02	0.376158	4
GOA	2.639E+02	7.884E-01	4.091E-01	2.639E+02	8.400E-03	2.639E+02	2.639E+02	0.822799	5
COOT	2.639E+02	7.885E-01	4.087E-01	2.639E+02	9.651E-03	2.639E+02	2.639E+02	0.388216	6
AVOA	2.639E+02	7.863E-01	4.150E-01	2.639E+02	2.154E-02	2.639E+02	2.640E+02	0.4107	7
HHO	2.639E+02	7.877E-01	4.109E-01	2.640E+02	8.407E-02	2.639E+02	2.641E+02	0.942015	8
DBO	2.639E+02	7.828E-01	4.251E-01	2.640E+02	1.217E-01	2.639E+02	2.642E+02	0.426548	9
ChOA	2.640E+02	7.832E-01	4.246E-01	2.640E+02	5.212E-02	2.640E+02	2.641E+02	0.476215	10
SCA	2.639E+02	7.892E-01	4.068E-01	2.641E+02	1.321E-01	2.639E+02	2.642E+02	0.375773	11
BO	2.640E+02	5.483E-01	1.133E-01	2.642E+02	1.889E-01	2.640E+02	2.645E+02	2.636269	12
SA	2.639E+02	7.960E-01	3.879E-01	2.644E+02	4.867E-01	2.639E+02	2.650E+02	0.703317	13
BWO	2.640E+02	7.938E-01	3.945E-01	2.646E+02	6.032E-01	2.640E+02	2.656E+02	0.466737	14
WOA	2.640E+02	7.792E-01	4.359E-01	2.650E+02	2.007E+00	2.640E+02	2.686E+02	0.37526	15

7. Conclusions

In this paper, we have presented the Hybrid COASaDE Optimizer, a novel combination of the Crayfish Optimization Algorithm (COA) and Self-adaptive Differential Evolution (SaDE) designed to address complex optimization challenges and solve engineering design problems. The proposed hybrid approach leverages COA's efficient exploration mechanisms, inspired by crayfish behavior, and SaDE's adaptive exploitation capabilities, characterized by its dynamic parameter adjustment. This synergy aims to balance the exploration and exploitation phases to enhance the algorithm's ability to effectively navigate diverse optimization landscapes. We have detailed the mathematical model of the Hybrid COASaDE algorithm, including the initialization and parameter setting, position update mechanisms, and integration of the mutation and crossover strategies from SaDE. The exploration phase utilizes COA's behavior-based strategy to avoid premature convergence, while the exploitation phase employs SaDE's adaptive techniques to refine the candidate solutions. Experimental evaluations were conducted using the CEC2022 and CEC2017 benchmark functions, demonstrating Hybrid COASaDE's superior performance compared to both traditional and state-of-the-art optimization algorithms. The results were analyzed through various methods, including convergence curves, search history plots, sensitivity analysis, and statistical analyses such as box plots and histograms. These analyses confirm the robustness and efficiency of Hybrid COASaDE in finding optimal solutions. Furthermore, the applicability of Hybrid COASaDE was validated through several engineering design problems, including the welded beam, pressure vessel, spring, speed reducer, cantilever, I-beam, and three-bar truss design problems. Hybrid COASaDE consistently outperformed the other optimizers on each design problem while achieving the optimal solution.

Author Contributions: Conceptualization, H.N.F., A.I. and S.N.M.; methodology, H.N.F., A.I. and S.N.M.; software, H.N.F., A.I. and S.N.M.; validation, M.A.A.-B. and M.A.; formal analysis, H.N.F., M.A.A.-B. and M.A.; investigation, H.N.F., M.A.A.-B. and M.A.; writing—original draft preparation, H.N.F., A.I., S.N.M., M.A.A.-B. and M.A.; writing—review and editing, H.N.F., A.I., S.N.M. and M.A.A.-B.; visualization, H.N.F. and S.N.M.; supervision, H.N.F. and S.N.M.; project administration, H.N.F.; funding acquisition, M.A.A.-B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ajman University.

Data Availability Statement: The original data presented in the study are openly available in the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abdel-Basset, M.; Abdel-Fatah, L.; Sangaiah, A.K. Metaheuristic algorithms: A comprehensive review. In *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*; Academic Press: Cambridge, MA, USA, 2018; pp. 185–231.
2. Kochenderfer, M.J.; Wheeler, T.A. *Algorithms for Optimization*; MIT Press: Cambridge, MA, USA, 2019.
3. Diwekar, U.M. *Introduction to Applied Optimization*; Springer Nature: Berlin/Heidelberg, Germany, 2020; Volume 22.
4. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
5. Fakhouri, H.N.; Alawadi, S.; Awaysheh, F.M.; Hamad, F. Novel hybrid success history intelligent optimizer with gaussian transformation: Application in CNN hyperparameter tuning. *Clust. Comput.* **2023**, *27*, 3717–3739. [CrossRef]
6. Li, Q.; Tai, C.; Weinan, E. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. *J. Mach. Learn. Res.* **2019**, *20*, 1474–1520.
7. Gill, P.E.; Murray, W.; Wright, M.H. *Practical Optimization*; SIAM: Philadelphia, PA, USA, 2019.
8. Yin, Z.Y.; Jin, Y.F.; Shen, J.S.; Hicher, P.Y. Optimization techniques for identifying soil parameters in geotechnical engineering: Comparative study and enhancement. *Int. J. Numer. Anal. Methods Geomech.* **2018**, *42*, 70–94. [CrossRef]
9. Guo, K.; Yang, Z.; Yu, C.H.; Buehler, M.J. Artificial intelligence and machine learning in design of mechanical materials. *Mater. Horizons* **2021**, *8*, 1153–1172. [CrossRef] [PubMed]
10. Fakhouri, S.N.; Hudaib, A.; Fakhouri, H.N. Enhanced optimizer algorithm and its application to software testing. *J. Exp. Theor. Artif. Intell.* **2020**, *32*, 885–907. [CrossRef]
11. Shukri, S.E.; Al-Sayyed, R.; Hudaib, A.; Mirjalili, S. Enhanced multi-verse optimizer for task scheduling in cloud computing environments. *Expert Syst. Appl.* **2021**, *168*, 114230. [CrossRef]
12. Hudaib, A.A.; Fakhouri, H.N. Supernova optimizer: A novel natural inspired meta-heuristic. *Mod. Appl. Sci.* **2018**, *12*, 32–50. [CrossRef]
13. Fakhouri, H.N.; Alawadi, S.; Awaysheh, F.M.; Hani, I.B.; Alkhalaileh, M.; Hamad, F. A comprehensive study on the role of machine learning in 5G security: Challenges, technologies, and solutions. *Electronics* **2023**, *12*, 4604. [CrossRef]
14. Zhan, Z.H.; Shi, L.; Tan, K.C.; Zhang, J. A survey on evolutionary computation for complex continuous optimization. *Artif. Intell. Rev.* **2022**, *55*, 59–110. [CrossRef]
15. Ahvanooey, M.T.; Li, Q.; Wu, M.; Wang, S. A survey of genetic programming and its applications. *KSII Trans. Internet Inf. Syst. (TIIS)* **2019**, *13*, 1765–1794.
16. Žilinskas, A.; Calvin, J. Bi-objective decision making in global optimization based on statistical models. *J. Glob. Optim.* **2019**, *74*, 599–609. [CrossRef]
17. Fakhouri, H.N.; Hudaib, A.; Sleit, A. Hybrid particle swarm optimization with sine cosine algorithm and nelder–mead simplex for solving engineering design problems. *Arab. J. Sci. Eng.* **2020**, *45*, 3091–3109. [CrossRef]
18. Lan, G. *First-Order and Stochastic Optimization Methods for Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 1.
19. Diakonikolas, I.; Kamath, G.; Kane, D.; Li, J.; Steinhardt, J.; Stewart, A. Sever: A robust meta-algorithm for stochastic optimization. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 1596–1606.
20. Fakhouri, H.N.; Hudaib, A.; Sleit, A. Multivector particle swarm optimization algorithm. *Soft Comput.* **2020**, *24*, 11695–11713. [CrossRef]
21. Erol, O.K.; Eksin, I. A new optimization method: Big bang–big crunch. *Adv. Eng. Softw.* **2006**, *37*, 106–111. [CrossRef]
22. Powell, W.B. A unified framework for stochastic optimization. *Eur. J. Oper. Res.* **2019**, *275*, 795–821. [CrossRef]
23. Bitar, R.; Wootters, M.; El Rouayheb, S. Stochastic gradient coding for straggler mitigation in distributed learning. *IEEE J. Sel. Areas Inf. Theory* **2020**, *1*, 277–291. [CrossRef]
24. Mathew, T.V. Genetic Algorithm. 2012. Available online: https://datajobs.com/data-science-repo/Genetic-Algorithm-Guide-[Tom-Mathew].pdf (accessed on 9 June 2024).
25. Zedadra, O.; Guerrieri, A.; Jouandeau, N.; Spezzano, G.; Seridi, H.; Fortino, G. Swarm intelligence-based algorithms within IoT-based systems: A review. *J. Parallel Distrib. Comput.* **2018**, *122*, 173–187. [CrossRef]
26. Dorigo, M.; Stützle, T. *Ant Colony Optimization: Overview and Recent Advances*; Springer: Berlin/Heidelberg, Germany, 2019.
27. Fakhouri, H.N.; Awaysheh, F.M.; Alawadi, S.; Alkhalaileh, M.; Hamad, F. Four vector intelligent metaheuristic for data optimization. *Computing* **2024**, *106*, 2321–2359. [CrossRef]
28. Yang, X.S. *Nature-Inspired Optimization Algorithms*; Academic Press: Cambridge, MA, USA, 2020.
29. Deng, W.; Xu, J.; Gao, X.Z.; Zhao, H. An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *52*, 1578–1587. [CrossRef]
30. Seyyedabbasi, A. WOASCALF: A new hybrid whale optimization algorithm based on sine cosine algorithm and levy flight to solve global optimization problems. *Adv. Eng. Softw.* **2022**, *173*, 103272. [CrossRef]

31. Ashraf, A.; Almazroi, A.A.; Bangyal, W.H.; Alqarni, M.A. Particle Swarm Optimization with New Initializing Technique to Solve Global Optimization Problems. *Intell. Autom. Soft Comput.* **2022**, *31*, 191. [CrossRef]
32. Che, Y.; He, D. A hybrid whale optimization with seagull algorithm for global optimization problems. *Math. Probl. Eng.* **2021**, *2021*, 6639671. [CrossRef]
33. Braik, M.; Al-Zoubi, H.; Ryalat, M.; Sheta, A.; Alzubi, O. Memory based hybrid crow search algorithm for solving numerical and constrained global optimization problems. *Artif. Intell. Rev.* **2023**, *56*, 27–99. [CrossRef]
34. Wang, Z.; Luo, Q.; Zhou, Y. Hybrid metaheuristic algorithm using butterfly and flower pollination base on mutualism mechanism for global optimization problems. *Eng. Comput.* **2021**, *37*, 3665–3698. [CrossRef]
35. Jia, H.; Zhou, X.; Zhang, J.; Abualigah, L.; Yildiz, A.R.; Hussien, A.G. Modified crayfish optimization algorithm for solving multiple engineering application problems. *Artif. Intell. Rev.* **2024**, *57*, 127. [CrossRef]
36. Daulat, H.; Varma, T.; Chauhan, K. Augmenting the Crayfish Optimization with Gaussian Distribution Parameter for Improved Optimization Efficiency. In Proceedings of the 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS), Tamil Nadu, India, 17–19 April 2024; pp. 462–470.
37. Jia, H.; Rao, H.; Wen, C.; Mirjalili, S. Crayfish optimization algorithm. *Artif. Intell. Rev.* **2023**, *56*, 1919–1979. [CrossRef]
38. Qin, A.K.; Suganthan, P.N. Self-adaptive differential evolution algorithm for numerical optimization. In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Scotland, UK, 2–5 September 2005; Volume 2, pp. 1785–1791.
39. Eltaeib, T.; Mahmood, A. Differential evolution: A survey and analysis. *Appl. Sci.* **2018**, *8*, 1945. [CrossRef]
40. Pepelyshev, A.; Zhigljavsky, A.; Žilinskas, A. Performance of global random search algorithms for large dimensions. *J. Glob. Optim.* **2018**, *71*, 57–71.
41. Ye, P.; Pan, G. Selecting the best quantity and variety of surrogates for an ensemble model. *Mathematics* **2020**, *8*, 1721. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.