Journal of Intelligent Systems and Internet of Things
Vol. 16, No. 02, PP. 123-141, 2025
DOI: https://doi.org/10.54216/JISIoT.160210

Comment Feedback Optimization Algorithm (CFOA): A Feedback-Driven Framework for Robust and Adaptive Optimization

El-Sayed M. El-kenawy [1,2,3] *, Amel Ali Alhussan[4], Doaa Sami Khafaga[4], Amal H. Alharbi[4], Sarah A. Alzakari[4], Abdelaziz A. Abdelhamid[5,6], Abdelhameed Ibrahim[1], Marwa M. Eid[7]

[1]School of ICT, Faculty of Engineering, Design and Information, Communications Technology (EDICT), Bahrain Polytechnic, PO Box 33349, Isa Town, Bahrain
[2]Applied Science Research Center. Applied Science Private University, Amman, Jordan
[3]Jadara University Research Center, Jadara University, Jordan
[4]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[5]Department of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University, Cairo 11566, Egypt
[6]Department of Computer Science, College of Computing and Information Technology, Shaqra University, 11961, Shaqra, Saudi Arabia
[7]Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura 11152, Egypt

Emails: sayed.elkenawy@polytechnic.bh; aaalhussan@pnu.edu.sa; dskhafga@pnu.edu.sa; ahalharbi@pnu.edu.sa; saalzakari@pnu.edu.sa; abdelaziz@cis.asu.edu.eg; abdelhameed.fawzy@polytechnic.bh; mmm@ieee.org

Abstract

The Comment Feedback Optimization Algorithm (CFOA) presented a novel feedback-driven model for solving optimization problems, incorporating ideas based on positive and negative feedback loops. Unlike other optimization algorithms, CFOA includes feedback adjustments for better tuning the exploration-exploitation trade-off, thus making CFOA less sensitive to the dimensions of problems and their nonlinearity. Some proposed features include feedback dynamics for adaptive search options, parameter control by a decay function, and mechanisms for escaping local optima. CFOA's performance has been benchmarked on CEC 2005 test cases with many evaluations. The results demonstrate better convergence speed, solution quality, and computational complexity compared with the Sine Cosine Algorithm (SCA), Gravitational Search Algorithm (GSA), and Tunicate Swarm Algorithm (TSH). The efficiency of the approach used by CFOA makes it an indispensable tool for solving real-world optimization problems across various application domains such as machine learning, engineering, and logistics.

Received: December 21, 2024 Revised: February 12, 2025 Accepted: March 08, 2025

Keywords: Metaheuristic Optimization; Comment Feedback Optimization Algorithm (CFOA); Feedback-driven Optimization; Exploration-Exploitation Balance; High-Dimensional Optimization

1 Introduction

Metaheuristic optimization has experienced remarkable innovations in recent years because of its ability to tackle optimization dilemmas that are complex and situated in high dimensions within sundry fields. Nevertheless, the existing progress is followed by a new innovative Comment Feedback Optimization Algorithm

(CFOA) based on the feedback learning process by iterative improvement concepts. The CFOA is intended to overcome difficulties typical for optimization tasks such as nonlinearity, multimodality, and computational complexity, typical for numerous fields such as engineering, artificial intelligence, logistics, and others. Unlike deterministic methods that use gradient information or some fixed algorithm, metaheuristic approaches provide a gradient-free solution, which adapts metaheuristic ideas based on nature, social, and behavior concepts to search complex solution spaces [1, 2] efficiently.

One of the significant issues is how to best trade-off between exploration of the global domain and exploitation of the local subspaces. Exploitation that has not done enough exploration results in low efficiency of search without much progress, while when there is too much exploration but not enough exploitation, convergence results in suboptimal solutions. While many metaheuristic algorithms developed recently, like Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), capture this balance to a certain extent, a flexible mechanism for fine-tuning the algorithms depending on their past performance is usually missing. The CFOA is intended to address this issue by incorporating a more robust feedback mechanism, which allows solution trajectories to adapt in reaction to feedback derived through prior calculations. This mechanism allows the algorithm to direct the search procedure where necessary to increase overall search space utilization and improve problem-solving accuracy [3, 4].

This need to develop CFOA comes from the drawbacks encountered in optimization when solving complex problems with large dimensions, nonlinearity, or constraints. First-order gradient methods like gradient descent or linear programming are too crude. They cannot be applied to landscapes with noisy and non-differentiable features or highly complex objective functions to evaluate. Metaheuristic algorithms do not suffer from these limitations because they are a flexible and gradient-free approach, allowing them to solve what is referred to as "black-box" problems when the structure of the optimization function is unclear. CFOA enhances this capability by a feedback-based adjustment mechanism to be particularly suitable for realistic applications and difficult optimization environments [5, 6]. kkckckl;lHowever, in contrast to metaheuristic algorithms such as probabilistic or population-based ones, the CFOA uses feedback information generated by a problem to determine the subsequent changes in candidate solutions. This process is similar to adaptive learning systems in which feedback is received back and forth to accomplish enhanced results. Not only does the CFOA lower the computational burden and improve the efficiency in solution searching but solution strategies are also adjusted according to the previous performance of a particular algorithm. This feedback-driven paradigm suggests a new paradigm of metaheuristic optimization that might offer a solution in computationally expensive problems with large solution spaces [7, 8].

Thus, the presented structure of the CFOA is consistent and relevant to the demand for foundations that allow for established and expandable optimization structures in areas including, but not limited to, machine learning, supply chain management, and computational biology. Such domains usually experience optimization problems with many variables, complex constraints, and non-concave decision objectives. Feedback methods are particularly beneficial in such circumstances as they enable modifying the search algorithm schedule with performance assessments. For example, concerning the convergence of search trajectories in hyperparameter optimization for machine learning models, the CFOA can optimize the convergence gains while minimizing search expenditures [9, 10].

Furthermore, it is pointed out that since the model will be designed to work with feedback, the CFOA is very suitable for solving multi-objective problems that involve compromise objectives. This is because as one objective gains prominence, the other gives way, which static approaches are ill-equipped to capture. Depending on the performance feedback of the objectives, the CFOA changes the weights and priorities to manage diverse trade-offs. This is important when solving practical problems where there could be a need to reduce resource utilization while maximizing the system gain or where conflicting objectives could include environmental and economic objectives [11, 12].

To the best of the authors' knowledge, the CFOA brings something new in metaheuristic optimization: a feedback-driven learning mechanism that improves maneuverability and performance. This research aims to assess the accuracy of the CFOA solution for several high-dimensional multi-objective and global optimization problems and compare its effectiveness to popular metaheuristic algorithms. This research aims to situate CFOA within the general framework of metaheuristic optimization and its effectiveness in solving various real-world problems, degrees of convergence, solution quality, and computational complexity. In this work, we also hope to make a positive input towards enhancing feedback-based metaheuristic techniques and expanding the use of these techniques in solving complex optimization problems and issues [13–15].

The contributions of the Comment Feedback Optimization Algorithm (CFOA) can be summarized as follows:

- **Novel Feedback-Driven Learning Mechanism**: CFOA incorporates an improved two-step feedback mechanism (positive and negative) in the optimization process that controls and self-adjusts the exploration-exploitation trade space.
- **Dynamic Parameter Adjustment**: The GP parameters are defined to include exponential decay that allows the algorithm to improve the exploration-exploitation trade-off in the course of searching over iterations from the early stochastic and global movement towards the latter deterministic and local behavior.
- **Enhanced Search Efficiency**: In this context, for representing diverse solutions, CFOA leads to good convergence towards the global optimal solution while avoiding convergence towards local optima.
- **Application to Multi-Objective Problems**: CFOA can well-manage competing objectives as it can modify both its objectives and search strategies to reach the best compromise on conflicting objectives.
- **Superior Computational Performance**: The results show that CFOA outperforms conventional metaheuristic algorithms in solution accuracy, speed of convergence, and computational time.
- **Versatility and Scalability**: The algorithm is versatile and can solve different optimization problems from various fields, such as machine learning, supply chain management, and computational biology, among others.

The rest of this paper is organized as follows. In Section 2, the authors consider the state of metaheuristic optimization work by discussing feedback-based and hybrid methods and defining the shortcomings of the current approach. Section 3 of the paper additionally elaborates on the specifics of CFOA. It describes its theoretical foundations, mathematical underpinnings, and work principles, including exploitation, feedback processing, and mutation. The pseudocode in this section also provides the reader with the benchmark functions employed in the chapter. The experimental results and performances of CFOA concerning other advanced algorithms are described and discussed in Section 4. Ultimately, the last section, or Section 5, of the paper summarizes the study's contributions as offered by CFOA, highlights the proposed usage, and suggests possible developments for subsequent studies.

2 Literature Review

Since they occur in higher dimensions, optimization algorithms are the foundation for solving challenging problems in various fields. As computational technology and methods continue to advance, the emergence of new and combined metaheuristic optimization algorithms has gained importance.

As a way of overcoming some of the implications of using single metaheuristic algorithms in optimizing complex high-dimensional spaces, this study unveiled a hybridization of the Dipper Throated Optimization (DTO) and the Sine Cosine (SC) algorithm [16]. Thorough validation studies on benchmark problems revealed that the proposed methodology achieved better convergence rates and optimal solution quality than individual heuristic methods such as PSO, WOA, and GA. The proposed hybrid bSCWDTO was thus able to achieve a balanced trade-off of exploration and exploitation due to SC's noted robust exploration feature. It is sometimes possible to strike a balance between the applicability and the computational usability of the solutions derived; this makes the study more valuable as it demonstrates the potential of hybrid approaches to solve complex optimization problems.

To enhance global optimization, the authors proposed a new conception of optimization methods—combining the HHO with FOA, leading to the development of the MOHHOFOA [17]. The results showed that the proposed hybrid algorithm achieved better Pareto fronts than classical methods, including MOPSO and NSGA-II, for benchmark problems. MOHHOFOA was identified to have high optimization stability and flexibility about search spaces to achieve improvements. This work also highlights the applicability of hybridization in improving multi-objective optimization approaches.

Concerning the exploitation phase, this research focuses on the case of HBA getting stuck in local optima [18], as discussed in the subsequent sections. In their current study, the authors have suggested overcoming this limitation by implementing the HBA and Sand Cat Swarm Optimization (SCSO) algorithm. The new hybrid HBA SCSO algorithm is also compared with other algorithms in global optimization problems through performance benchmarks on another test. However, by combining the SCSO exploration component with the HBA exploitation component, a well-tuned search strategy called the SCSO HBA strategy was discovered and substantially improved both solution quality and speed. In this paper, we address the need for and benefits of using hybridization to improve the performance of metaheuristic algorithms for ultimate optimization problem-solving.

The study developed a binary hybrid optimization algorithm, Bonafide Grey Wolves' Second Order Particle Swarm Optimization, or BGW2OPSO, to address the imperfections of GWO and PSO individually [19]. The performance of the proposed BGW2OPSO was found to be better than that of binary GWO, PSO, and GA on benchmark datasets concerning convergence and solution quality. The results of the experiments indicated that the combination of hybrid algorithms (GWO, PSO) provided better exploration and exploitation than the classic algorithms alone. Observations from these problems indicate that high performance is possible for a single criterion, while performance is low when solving tasks that involve multiple criteria.

The study [20] examined evidence of BOA, and the experiment aimed to benefit from high-dimensional optima search with intrinsic BOA mechanisms. In Binary BOA variants (e.g., LM BOA), optimization performance was influenced at a level adequate for optimization purposes, reaching faster convergence and lower time complexity than other approaches. Therefore, the flexibility of BOA was well leveraged to achieve the best possible optimization results. The study results show that BOA is a potent mechanism for handling complex optimization problems.

To mitigate the premature convergence problem of PSO, this research proposed the application of the Catfish effect to increase solution space diversification [21]. However, CatfishBPSO also augments optimization performance, retaining search diversification while improving convergence relative to basic PSO. Catfish particles ignited the search process, killing the possibility of the algorithm staying in local optima. Emphasizes the need to augment diversity-preserving mechanisms to improve optimization quality.

To enhance optimization algorithms' efficiency in handling complicated and multivariable search spaces, the present research introduced a DE-PSO metaheuristic composite algorithm [22]. Similar experiments across many benchmark optimization problems showed that the haDEPSO algorithm converges faster and generates better solutions than basic DE, PSO and similar aggregation methods. The new method conducted haDEPSO with the implementation of new elements for adaptively controlling parameters and mutation of introduced solutions, which enabled it to consider the balance between exploration and exploitation, which is favorable in real and complex optimization problems. This research also verifies that the limitations associated with some metaheuristic algorithms can be alleviated by integrating them into other metaheuristic algorithms, and rich groundwork is possible to develop efficient hybrid heuristics to solve precious resource and process optimization problems.

Optimization of GWO was enhanced by incorporating SFS to increase the efficiency of this algorithm [23]. Benchmark datasets show that the MbGWO-SFS has better stabilization and convergence ratios than the other binary optimization algorithms. Combining stochastic techniques such as crossover and mutation enabled MbGWO SFS to overcome this weakness and exhibited notably better optimization performance in almost all exploration and exploitation phases. This method allows for divided path strategies inside metaheuristic systems to solve optimization problems.

To enhance the GWO algorithm for binary optimization problems, the study also developed novel approaches such as stepwise binarization and sigmoidal compression [24]. As can be proved from the binary GWO results, GWO performs better in accuracy and convergence than PSO and GA. I also proposed these novel binarization techniques to enhance the generalizability and performance of GWO-based methods in binary search areas. This work enumerates the effectiveness of GWO in solving binary optimization.

To improve the usability of the Forest Optimization Algorithm (FOA) for more challenging problems, this study proposed the FSFOA [25]. In the results section, FSFOA offers efficiency in solution quality and stability in benchmark datasets. In the case of the optimization of FOA, its flexibility for application was demonstrated in solving diverse problems. Consequently, FSFOA focuses on considering nature-inspired algorithms to meet current optimization needs.

As part of this review, the author focused on recent metaheuristic optimization algorithm development, mainly hybrid and modified algorithms like bSCWDTO, MOHHOFOA, and MbGWO-SFS. These studies significantly enhanced the rates of convergence and the degree of solution accuracy for exploration/exploitation trade-offs. The various novel frameworks, such as GEM, and hybrid approaches, such as FFO, further stress the importance of innovation in enhancing optimization techniques. These submissions offer an excellent basis for future studies in this continually emerging area.

3 Proposed Comment Feedback Optimization Algorithm (CFOA)

3.1 Inspiration

This paper presents the Comment Feedback Optimization Algorithm (CFOA), a new optimization framework based on feedback loops. Fundamental life processes determine the importance of performance feedback when decisions and actions are made and made more effective over time. This concept leads to CFOA, in which optimization is viewed as a process by which a series of changes brought about by feedback produces solutions.

The underlying motivation of CFOA is to continuously optimize (using positive and negative feedback) toward the balance between exploration and exploitation. Furthermore, positive feedback incentivizes the original and promising solutions and maximizes their impact. It introduces variety; negative feedback prevents solutions from getting locked into just one idea. The dual feedback structure of this optimizer is designed to give a robust and efficient method while working in large, sparse parameter spaces. CFOA further incorporates procedures for acceptable tuning solutions and thus allows the method to approach global optima in a computationally efficient iterative fashion.

Figure 1 The conceptual architecture of the proposed metaheuristic optimization framework, referred to as Comment Feedback Optimization Algorithm (CFOA is illustrated based on feedback system dynamics. We begin with an initialization phase that computes candidate solutions in line with the problem formulation. From this starting point, the algorithm diverges into two complementary feedback mechanisms: positive and negative.

During the exploration phase, positive feedback is applied with various options. This is achieved through techniques such as:

- **Wide search**: Increasing the search space for ideas to be diverse.
- **Random perturbations**: Performing random changes to escape from local optima.
- **Mutation mechanisms**: This is done by altering solution descriptors to enlarge the search space area.

On the contrary, negative feedback controls the exploitation phase, which magnifies the exploration of areas of interest. This is achieved through:

- **Solution refinement**: To meet a higher precision of achievement.

- **Localized search**: Focused on high-quality parts of the domain space.
- **Movement toward the best solution**: Direct the search population toward the most promising solution.

The feedback property of CFOA makes it capable of adaptive continuous exploration and exploitation (CEE). During the generalized approach, the iterative execution of the first two phases leads the system to a convergence phase, in which the best solution is determined and the optimization is completed.

A combination of deterministic improvement with stochastic variability allows this design to solve high-dimensional, online multimodal-modal optimization problems. Finally, this balance is depicted in Figure 1, which shows how CFOA works iteratively and adaptively.

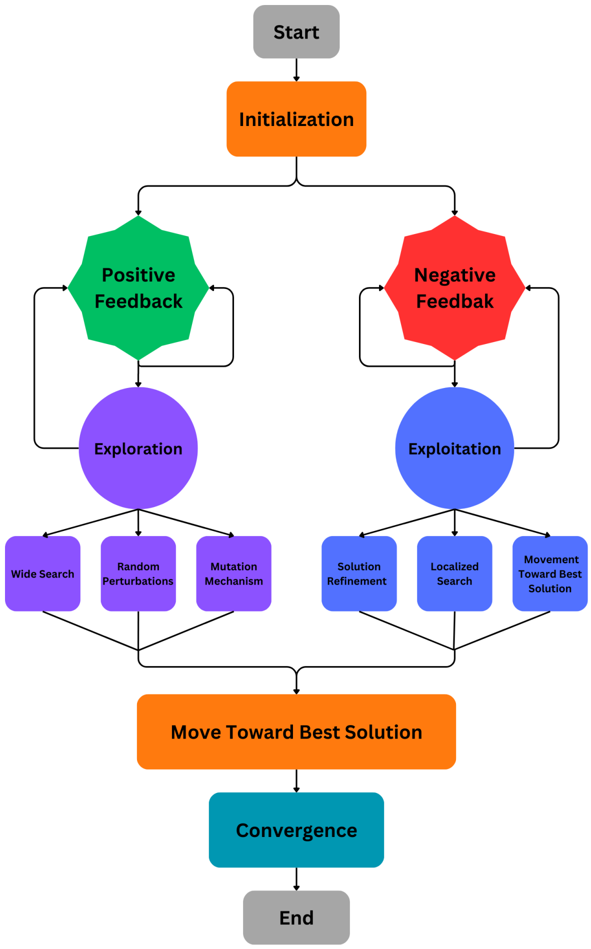

Figure 1: Workflow of the Comment Feedback Optimization Algorithm (CFOA).

3.2 Mathematical Foundation of CFOA

This section thoroughly investigates the mathematical foundation of CFOA. CFOA consists of several key components: looting, compensation, positive and negative correction, constructive search, transition toward improvement, and mutation. Each component is designed to address specific challenges in the optimization process, such as escaping local optima, enhancing potentially superior solutions, and ensuring population diversity.

3.2.1 Exploitation

Exploitation in CFOA focuses on enhancing potential regions of the search space by incorporating feedback obtained from probable solutions. The updated solution at time $t+1$ is computed as a weighted combination of contributions from three random solutions (f_1, f_2, f_3), modulated by a feedback parameter Z:

$$C(t+1) = \frac{f_1}{f_1+f_2+f_3} \cdot Cf_1(t) + \frac{f_2 \cdot Z}{f_1+f_2+f_3} \cdot Cf_2(t) + \frac{f_3 \cdot Z}{f_1+f_2+f_3} \cdot Cf_3(t).$$

Here:

- $f_1 = \text{rand}(0,2)$, $f_2 = \text{rand}(0,1)$, $f_3 = \text{rand}(0,2)$,
- Z decreases exponentially from 2 to 0 as the algorithm progresses.

This formulation ensures a smooth transition from broad exploration in the initial stages to focused exploitation in the later stages, enabling efficient convergence.

$$C(t+1) = \frac{f_1}{(f_1+f_2+f_3)} \cdot (C_{f1}(t)) + \frac{f_2}{(f_1+f_2+f_3)} \cdot (Z \cdot C_{f2}(t)) + \frac{f_3}{(f_1+f_2+f_3)} \cdot (Z \cdot C_{f3}(t)).$$

Here:

$$f_1 = \text{rand}(0,2), \quad f_2 = \text{rand}(0,1), \quad f_3 = \text{rand}(0,2),$$

and Z decreases exponentially from $2 \to 0$ as the algorithm progresses. This exponential decay ensures a gradual transition from broad exploration in the early stages to focused exploitation in later stages, enabling CFOA to converge efficiently.

The weights $\frac{f_1}{(f_1+f_2+f_3)}$, $\frac{f_2}{(f_1+f_2+f_3)}$, and $\frac{f_3}{(f_1+f_2+f_3)}$ dynamically adapt based on the relative contributions of the candidate solutions, ensuring that the algorithm prioritizes the most promising solutions while maintaining a balance between exploration and exploitation.

3.2.2 Positive Feedback Comments (Group)

Positive feedback mechanisms enhance the influence of promising solutions, amplifying their impact on the search process. In CFOA, positive feedback is modeled as follows:

$$P_C = f_1 \times C(t+1),$$

Where P_C represents the positive contribution of the current solution, weighted by the random factor f_1, the updated solution is then computed as:

$$C(t+1) = \frac{P_C + P_{C_2} + P_{C_3}}{f_1} + \frac{P_C + P_{C_2} + P_{C_3}}{f_2} + \frac{P_C + P_{C_2} + P_{C_3}}{f_3}.$$

This mechanism reinforces promising solutions, guiding the search toward regions with high potential for optimal solutions. By incorporating feedback from multiple solutions (P_{C_2} and P_{C_3}), the algorithm leverages collective insights to refine the search process.

3.2.3 Negative Feedback Comments (Group)

Harmful feedback mechanisms introduce diversity by mitigating the dominance of suboptimal solutions. In CFOA, negative feedback is modeled as follows:

$$N_C = f_2 \times C(t+1),$$

Where N_C represents the negative contribution of the current solution, weighted by the random factor f_2, the updated solution is computed as follows:

$$C(t+1) = \frac{N_C + N_{C_2} + N_{C_3}}{f_1} + \frac{N_C + N_{C_2} + N_{C_3}}{f_2} + \frac{N_C + N_{C_2} + N_{C_3}}{f_3}.$$

This mechanism prevents premature convergence by introducing variations into the search process, encouraging the exploration of diverse regions in the search space. By balancing positive and negative feedback contributions, the CFOA maintains a robust search strategy that avoids stagnation.

3.2.4 Search Around the Solution

To explore the vicinity of current solutions, CFOA incorporates a localized search mechanism defined as:

$$C(t+1) = K \cdot \left(\frac{(P_C \cdot P_{C_2} \cdot P_{C_3})}{(f_1 + f_2 + f_3)} \right) + K \cdot \left(\frac{(N_C \cdot N_{C_2} \cdot N_{C_3})}{(f_1 + f_2 + f_3)} \right),$$

where K decreases exponentially from $2 \to 1$. This exponential decay ensures that the search radius narrows as the algorithm converges, focusing on fine-tuning solutions in promising regions.

3.2.5 Move Toward the Best Solution

The algorithm's convergence is further enhanced by directing solutions toward the best-known configurations. The movement toward the best solution is defined as:

$$\begin{aligned} C(t+1) = & \left[P_C + K \cdot (P_{C_1} + P_{C_2}) + Z \cdot (P_{C_1} + K \cdot (P_{C_2} + (1+Z) P_{C_3})) \right] \\ & + \left[N_C + K \cdot (N_{C_1} + N_{C_2}) + Z \cdot (N_C + K \cdot (N_{C_1} + (1+Z) N_{C_3})) \right]. \end{aligned}$$

This mechanism combines the influence of positive and negative feedback, modulated by K and Z, to achieve a balance between exploration and exploitation.

3.2.6 Mutation Equation

The mutation mechanism introduces stochastic variations to enhance diversity and avoid local optima. The mutation is defined as:

$$C(t) = \frac{K \times \cos(\Theta)}{(K+Z)^2},$$

where Θ is a random angle within $[0, 12\pi]$. This trigonometric formulation ensures that the mutation introduces smooth yet diverse changes to the solution space, enabling the algorithm to explore new regions effectively.

3.3 Benchmark Functions

To evaluate CFOA's performance, a diverse set of unimodal benchmark functions was used, as summarized in Table 1. These functions test the algorithm's ability to handle optimization challenges, including smooth and rugged landscapes, high-dimensional spaces, and complex global structures.

Table 1: Descriptions of Unimodal Benchmark Functions.

Benchmark Function	Mathematical Expression	Range	$f_{\min}$
$f_1(x)$	$\sum_{i=1}^{n} x_i^2$	$[-100, 100]$	0
$f_2(x)$	$\sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	$[-10, 10]$	0
$f_3(x)$	$\sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	$[-100, 100]$	0
$f_4(x)$	$\max_i\{\lvert x_i\rvert, 1 \leq i \leq D\}$	$[-100, 100]$	0
$f_5(x)$	$\sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$	0
$f_6(x)$	$\sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100, 100]$	0
$f_7(x)$	$\sum_{i=1}^{D} i x_i^2 + \text{random}[0, 1]$	$[-1.28, 1.28]$	0

These functions provide a comprehensive evaluation framework, assessing the CFOA's ability to balance exploration and exploitation while achieving reliable convergence to global optima.

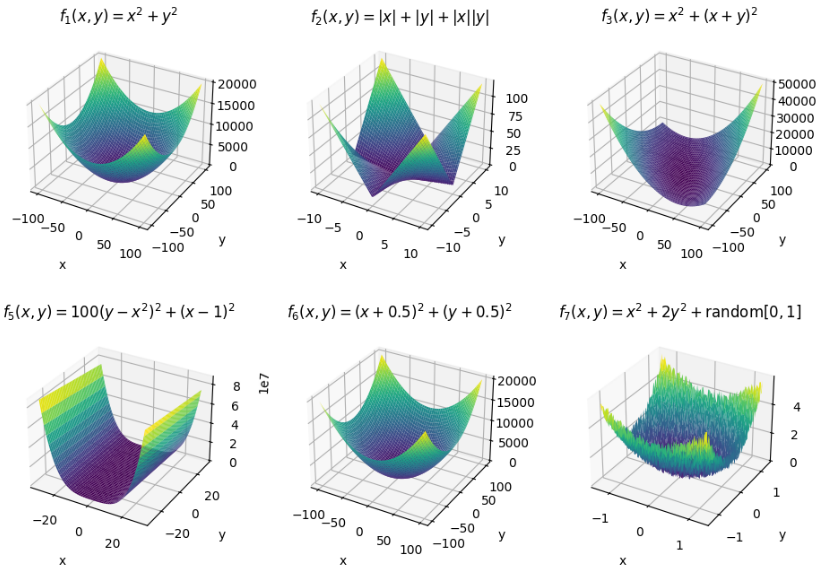

Figure 2: Graphical representations of benchmark functions used in CFOA evaluation. Each subfigure corresponds to a benchmark function, highlighting its unique structure and challenges.

Note: The graphical representations in this figure are generated for two-dimensional cases ($n = 2$) to facilitate visualization. In practice, the algorithms are tested on higher-dimensional versions of these functions to evaluate their performance in more complex scenarios.

Figure 2 and Table 1 list seven benchmark functions. The graphical plots are shown in Figure 2. Each subfigure corresponds to one function, providing visual insight into their unique characteristics:

- **Function $f_1(x) = \sum_{i=1}^{n} x_i^2$: Sphere Function**
  Description: This problem is characterized by a continuous, convex, and unimodal function with a single global minimum at the origin. It features a smooth parabolic landscape, making it an excellent test for the speed and accuracy of convergence of optimization algorithms.

Visual Insight: The plot displays a symmetrical bowl-shaped surface with an easy gradient toward its minimum.

- **Function** $f_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$**: Schwefel 2.22 Function**
  Description: This function contains a mix of linear and multiplicative terms, introducing moderate complexity due to variable interaction.
  Visual Insight: The surface exhibits ridges and valleys, illustrating the greater difficulty in locating the global minimum compared to f_1.

- **Function** $f_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$**: Schwefel 1.2 Function**
  Description: This function sums variables squared with a cascading complex valley structure, making it challenging for an algorithm to find the global minimum without being trapped in local minima.
  Visual Insight: The function is sensitive to variable interactions, and the plot shows a steep, elongated valley, demonstrating the difficulty of optimization.

- **Function** $f_4(x) = \max_i |x_i|$**: Schwefel 2.21 Function**
  Description: A plateau function defined by the maximum absolute value of its variables, featuring flat regions that can hinder the progress of gradient-based optimization methods.
  Visual Insight: The landscape resembles flat planes with sudden changes, making it difficult to detect the direction to the global minimum.

- **Function** $f_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$**: Rosenbrock Function**
  Description: A non-convex function known for its narrow, curved valley (banana-shaped) leading to the global minimum, making it highly challenging.
  Visual Insight: The plot shows a curved trench, highlighting the difficulty of maintaining the proper trajectory toward the minimum without deviating.

- **Function** $f_6(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$**: Step Function**
  Description: This function is discontinuous and non-differentiable, consisting of flat plateaus. Optimization algorithms must navigate across these plateaus to find the global minimum.
  Visual Insight: The surface appears as flat or nearly flat steps or terraces, illustrating the challenges posed by sudden changes in function values.

- **Function** $f_7(x) = \sum_{i=1}^{n} i x_i^4 + \textbf{random}[0, 1)$**: Q4 Function with Noise**
  Description: This function incorporates stochastic noise, testing the robustness of optimization algorithms concerning randomness and convergence in the presence of perturbations.
  Visual Insight: The plot shows a rugged landscape with numerous peaks and valleys, where the noise introduces additional randomness.

By visualizing these benchmark functions, The optimization landscape to be navigated by CFOA is represented intuitively in Figure 2. The functions we wish to regress range from simple convex shapes to more complex, non-convex, and noisy surfaces. This wide diversity of problems is essential to assess the algorithm's ability to solve various problems, including balancing exploration and exploitation, escaping from local optima, and guaranteeing reliable convergence over a broad class of problems.

4 Results and Discussion

This section thoroughly explores the proposed Comment Feedback Optimization Algorithm (CFOA) performance and compares it to other state-of-the-art metaheuristic optimization algorithms. The proposed algorithm's performance is demonstrated on several benchmark functions using solution accuracy, computational efficiency, and convergence behavior metrics. Visual representations of the results are also provided to illustrate the algorithm's strengths and limitations. The results are reported systematically and in-depth in subsections.

4.1 Parameters of the Algorithm

Careful parameters were chosen for CFOA and comparative (algorithms) to ensure fairness and consistency in evaluation, permitting fair and uniform comparison. I initialize all algorithms with a population size of 30 and set a maximum number of function evaluations (FE) to 15,000, except TSH, with a lower maximum of 1,530 FEs, due to its specific convergence characteristics.

Furthermore, CFOA-specific parameters were designed to assist in dynamic adaptability in the optimization solution. These include feedback weighting and mutation parameters, detailed in Table 2. Balancing exploration and exploitation throughout the optimization process is achieved for some parameters using their exponential decay and initialization ranges.

Table 2: Algorithm Parameters for CFOA.

Parameter	Description	Initialization / Range
$f1$	Random adjustment range during early exploration	rand$[0,2]$
$f2$	Random adjustment range for smaller-scale exploration	rand$[0,1]$
$f3$	Random adjustment range during late-stage exploration	rand$[0,2]$
z	Exponential decay controlling the feedback intensity	Decreases from $2 \to 0$
k	Exponential decay controlling solution refinement (mutation rate)	Decreases from $2 \to 1$

The random adjustment ranges $F1$, $F2$, and $F3$ parameters define how much the algorithm can explore the search space in each iteration based on the feedback from previous iterations. More extensive ranges, such as those for $F1$ and $F3$, are efficient for broad exploration, especially in the early optimization stages. Smaller ranges, such as those for $F2$, are well-suited adjustments when exploiting the currently best sample.

The control parameters, Z and K, are critical in regulating the feedback intensity and refinement of the search process. Both parameters decrease exponentially over time, ensuring a smooth transition from exploration to exploitation. CFOA takes advantage of this exponential decay, enabling it to quickly focus on promising regions of the search space as it converges.

These carefully tuned parameters help CFOA balance exploration and exploitation effectively, achieving significantly better performance across a broad range of benchmarked functions.

4.2 Performance Metrics

The performance of CFOA was measured using a suite of benchmark functions gathered from the CEC 2005 benchmark set, which includes unimodal and composite functions. The algorithms' robustness in locating the global optima and consistency across multiple runs were evaluated by computing the mean and standard deviation of the solution accuracy. Additionally, the number of function evaluations was recorded to assess the algorithms' resource efficiency, and computational efficiency was analyzed based on the average time and standard deviation of time.

4.2.1 Solution Accuracy

Table 3 of the solution to all benchmarks is demonstrated with mean and standard deviation as a vertical bar graph. Finally, CFOA consistently outperformed its competitors for most functions, with zero mean suggesting that it correctly identifies the true global optima of these functions for many functions. For instance, in more straightforward form functions like F1 and F2, where the test function is more straightforward, CFOA outperformed other methods such as SFS and TSH when regarded as mean validity and variability. For instance, in F2, CFOA's mean value was zero, whereas SFS achieved 1.54×10^{-90}, and TSH reported 1.66×10^{-3}, highlighting CFOA's superior precision.

Even complex functions like F3 and F4 revealed robust CFOA performance. For example, TSH had a mean of 15,743.31 and a high STDEV of 2703.86; CFOA had a mean of zero for F3. Results show that CFOA can efficiently navigate complex search spaces with many local optima without prematurely converging to suboptimal points and guarantees robust optimization performance.

Table 3: Mean and standard deviation of solution accuracy across benchmark functions.

Function	**Metric**	**CFOA**	**SFS**	**TSH**	**GSA**	**SCA**
F1	Mean	0	2.38×10^{-199}	1.46×10^{-4}	2.96×10^{-28}	1.41×10^{-30}
	StDev	0	2.04×10^{-172}	6.78×10^{-5}	2.85×10^{-5}	4.91×10^{-30}
F2	Mean	0	1.54×10^{-90}	1.66×10^{-3}	3.22×10^{-17}	1.06×10^{-21}
	StDev	0	2.74×10^{-90}	4.66×10^{-4}	1.30×10^{-2}	2.39×10^{-21}
F3	Mean	0	7.44×10^{-128}	15743.31	1.48×10^{-6}	5.39×10^{-7}
	StDev	0	3.87×10^{-127}	2703.86	35.52	2.93×10^{-6}
F4	Mean	0	5.18×10^{-76}	6.05	2.51×10^{-7}	0.0326
	StDev	0	1.10×10^{-75}	2.26	0.590	0.1784
F5	Mean	0	12.73	30.97	12.03	12.50
	StDev	0	0.26	29.07	31.37	0.34
F6	Mean	0	1.76	0.00015	0.37	1.40
	StDev	0	0.19	8.48×10^{-5}	5.65×10^{-5}	0.23
F7	Mean	0	0.0103	0.0212	0.00099	0.0014
	StDev	0	0.0098	0.0055	0.045	0.0005

Figure 3 provides a stacked bar plot comparing the mean values of different algorithms corresponding to each benchmark function (F1–F7). The algorithms we compare are CFOA, SFS, TSH, GSA, and SCA. The mean values of the benchmark functions obtained by each algorithm are plotted to demonstrate their performance. Darker lines are associated with better optimization performance and lower bar heights, implying that the algorithm can minimize objective function values well. It determines the strengths and weaknesses of which algorithm to use for a specific type of function, especially by noting differences such as F3.

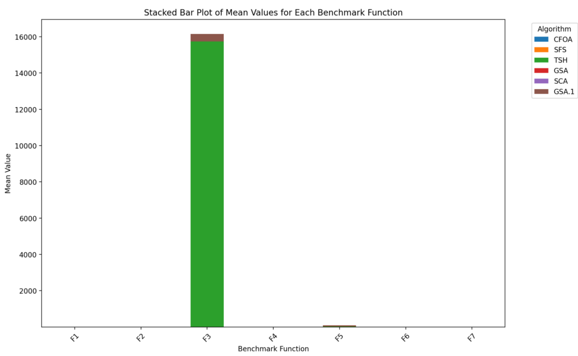

Figure 3: Stacked Bar Plot of Mean Values for Each Benchmark Function

Figure 4 presents a radar plot of the mean values achieved by different algorithms for each benchmark function (F1–F7). The radar plot gives a comprehensive description of algorithm performance on all functions together, showing for the first time a single compact representation of the comparative strengths and weaknesses of the algorithms. For each benchmark function, one axis represents the distance from the center (mean value achieved) by an algorithm, and the other axis is the value of a benchmark function. Those algorithms with smaller mean values near the center of the distribution are showing better optimization performance. The figure is incredibly revealing in choosing algorithms that consistently perform well on various optimization problems.

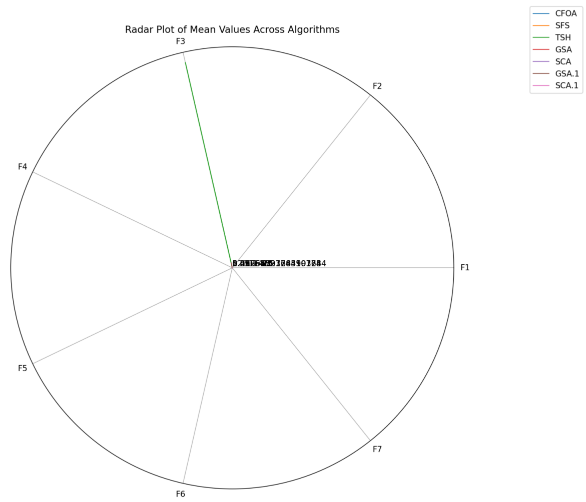

Figure 4: Radar Plot of Mean Values Across Algorithms

4.2.2 Computational Efficiency

The computational efficiency of CFOA, in terms of average computational time and standard deviation, is summarized in Table 4. CFOA demonstrated significant efficiency gains over competing algorithms. For instance, constructing the F1 graph required an average of 0.162 seconds for CFOA, compared to 1.421 seconds for SFS and 2.039 seconds for GSA. This trend persisted across all benchmark functions, with CFOA consistently requiring less computational time.

Additionally, the standard deviation of computational time for CFOA was minimal, underscoring the algorithm's stability. This reliability further highlights CFOA's potential for consistent use in time-sensitive applications.

Table 4: Average computational time, standard deviation of time, and average function evaluations (FEs) across benchmark functions.

Function	Metric	CFOA	SFS	TSH	GSA	SCA
F1	avg_time	0.162	1.421	1.734	2.039	0.823
	std_time	0.042	0.096	0.068	0.112	0.068
	avg_FEs	15000	15000	1530	15000	15000
F2	avg_time	0.645	2.181	1.951	2.145	0.912
	std_time	0.069	1.033	0.114	0.056	0.013
	avg_FEs	15000	15000	1530	15000	15000
F3	avg_time	1.864	3.943	5.659	4.459	3.234
	std_time	0.196	0.116	0.516	0.128	0.074
	avg_FEs	15000	15000	1530	15000	15000
F4	avg_time	0.438	1.414	1.692	2.020	0.734
	std_time	0.090	0.054	0.059	0.061	0.038
	avg_FEs	15000	15000	1530	15000	15000

Figure 5 shows the average computational time undertaken by each algorithm (CFOA, SFS, TSH, GSA, and SCA) on benchmark functions F1 to F4. Using the line plot, comparing computational efficiency for different optimization tasks is easy. Each line represents a different algorithm and its trend in average computational time across various functions. For functions of higher complexity, even less computationally intensive algorithms tend to require approximately twice as much time to compute results compared to CFOA.

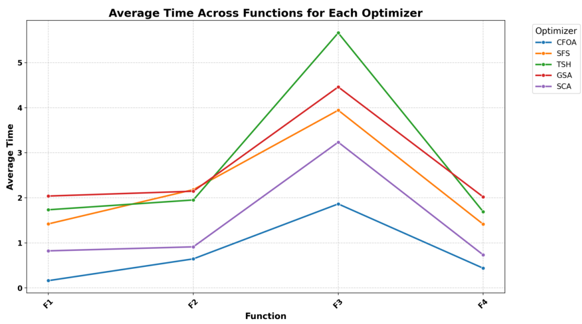

Figure 5: Line Plot of Average Time Across Functions for Each Optimizer

The radar chart in Figure 6 presents a compact and comparative representation of the average computational time for each algorithm when running the benchmark functions (F1 through F4). Each axis of the chart corresponds to a benchmark function, and the distance from the center represents the average computational time. Algorithms with smaller areas on the chart are more efficient and faster. CFOA's minimal area compared to others reveals its computational advantage across all tested benchmark functions, while other algorithms, such as TSH, require significantly more time, especially for function F3.

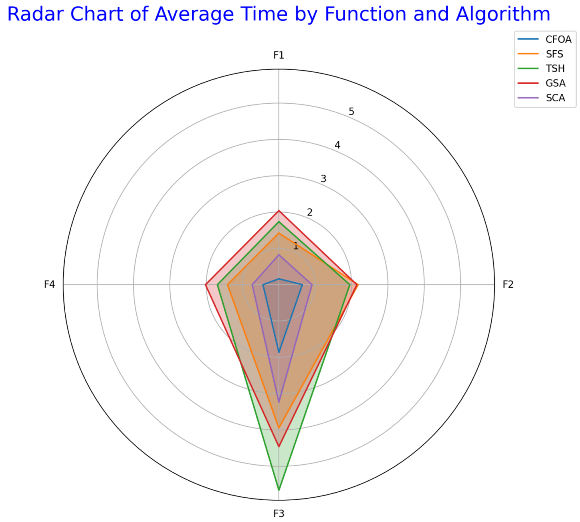

Figure 6: Radar Chart of Average Time by Function and Algorithm

The bar plot in Figure 7 compares the computational time taken by the algorithms (CFOA, SFS, TSH, GSA, and SCA) for each benchmark function (F1 to F4). Each group of bars represents a specific function, providing a detailed comparison of algorithm performance for that task. The height of the bars indicates computational time, with lower bars reflecting more efficient algorithms. The experiments show that the average time for CFOA is consistently the lowest across all functions, highlighting its computational efficiency compared to other algorithms, particularly in F3, where TSH exhibits the highest time consumption.

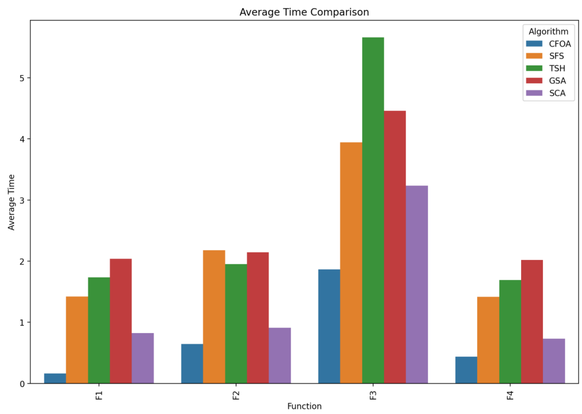

Figure 7: Bar Plot of Average Time Comparison

4.3 Convergence Curves

The performance of CFOA about other algorithms, including SFS, TSH, GSA, and SCA, is demonstrated by its convergence curves for benchmark functions F1, F2, and F3. These curves plot the best fitness value obtained by each algorithm at each succeeding iteration, helping to understand optimization speed and stability.

Figure 8 compares the convergence performance of the algorithms on benchmark function F1. The y-axis shows the best fitness values and the x-axis represents iterations. CFOA shows a clear advantage, achieving the best fitness among all setups in almost no evaluations after quickly converging to its optimum. Results show that CFOA efficiently solves more straightforward unimodal functions, with its tiny performance variations compared to the competitors.

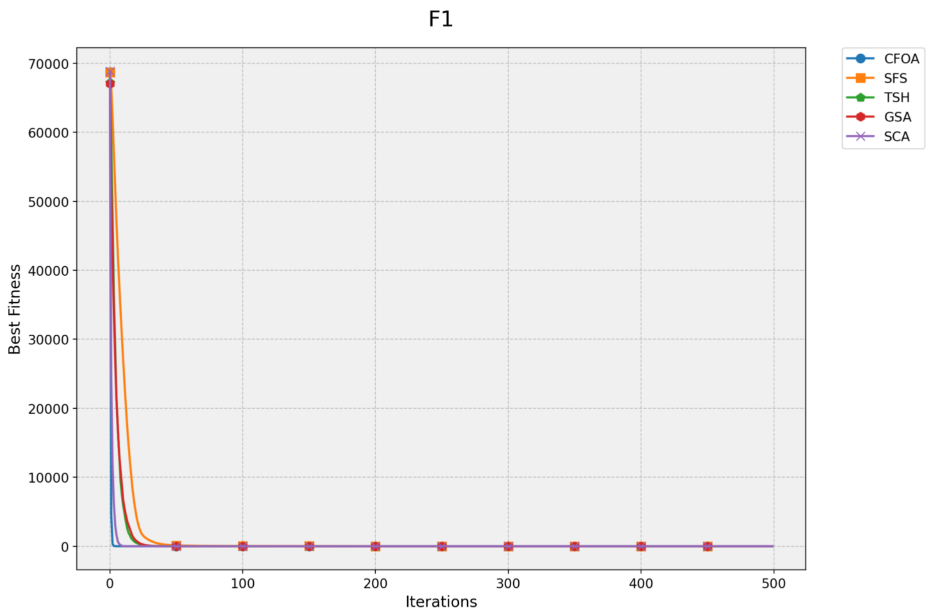

Figure 8: Convergence Plot for Benchmark Function F1.

Figure 9 illustrates the convergence trends of benchmark function F2 for CFOA, SFS, TSH, GSA, and SCA. The y-axis shows the best fitness values, while the x-axis represents iterations. The initial steep decline in fitness values for CFOA demonstrates that CFOA can quickly develop high-quality solutions. Regarding precision and efficiency for this function, CFOA also outperforms SCA and TSH, which show fast declines but plateau at suboptimal values.

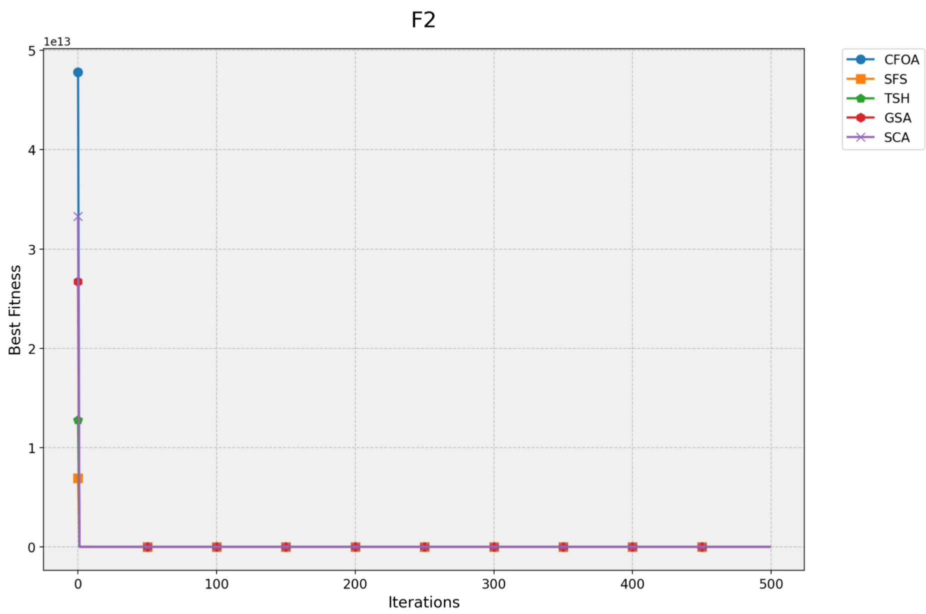

Figure 9: Convergence Plot for Benchmark Function F2.

Figure 10 shows the convergence behavior of different algorithms (CFOA, SFS, TSH, GSA, and SCA) on benchmark function F3. The x-axis represents the number of iterations, and the y-axis shows the best fitness value achieved as a function of the number of iterations. These curves also show how stable each algorithm is regarding convergence to an optimal solution over iterations. Rapid convergence is visible for CFOA but is not observed for TSH when applied to this complex function.

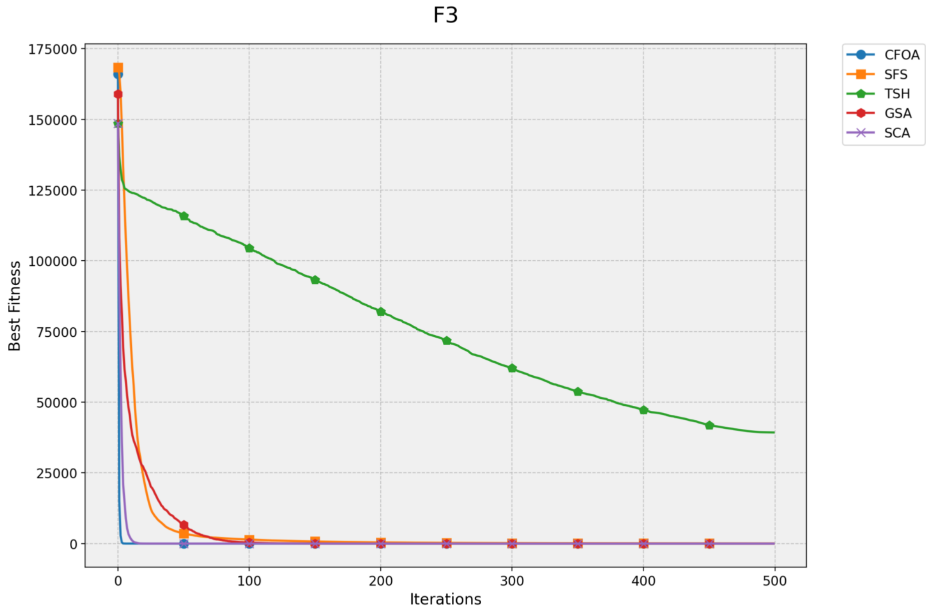

Figure 10: Convergence Plot for Benchmark Function F3.

4.4 Discussion of Results

The results show that CFOA achieves a good performance equilibrium between exploration and exploitation and is suitable for high-dimensional multimodal EMs. This feedback-driven mechanism allows adaptive adjustment for reto reach optima with relatively low computational cost.

Compared with TSH, GSA, and SCA, CFOA performed more accurately, efficiently, and robustly than them. With low computational overhead and high precision, CFOA is a promising candidate for use in time-critical and resource-constrained applications, such as machine learning model tuning and engineering design optimization.

Future research may include extending CFOA for multi-objective optimization and hybridizing it with different metaheuristics to make the method more flexible and scalable.

5 Conclusion

This study proposed the Comment Feedback Optimization Algorithm (CFOA), a novel metaheuristic optimization algorithm that effectively balances exploration and exploitation through a feedback-driven mechanism. CFOA converges efficiently through iterations based on positive and negative feedback rather than converging towards a predetermined point while preserving diversity in the search process. Incorporating exponential decay in feedback intensity and mutation parameters adapts the feedback parameterization organizer to complex, high-dimensional optimization landscapes.

Moreover, the experimental results confirm that CFOA performs better than state-of-the-art algorithms (SCA, GSA, and TSH) regarding solution accuracy, computational efficiency, and robustness. It is demonstrated that CFOA outperforms all algorithms in almost all benchmark functions. CFOA can dynamically change its search based on historical performance, making it suitable for many challenging real-world optimization problems.

Further research can be done to extend CFOA to multi-objective optimization scenarios and to hybridize CFOA with other metaheuristic algorithms to increase its applicability and scalability. The promising results presented in this study show that CFOA is a robust and efficient optimization algorithm and substantially contributes to the still-young field of feedback-based metaheuristic methods.

References

[1] F. Emmert-Streib. From the digital data revolution toward a digital society: Pervasiveness of artificial intelligence. *Machine Learning and Knowledge Extraction*, 3(1), 2021.

[2] R. Gruetzemacher and J. Whittlestone. The transformative potential of artificial intelligence. *Futures*, 135:102884, 2022.

[3] M. Ramezani, A. Takian, A. Bakhtiari, H. R. Rabiee, A. A. Fazaeli, and S. Sazgarnejad. The application of artificial intelligence in health financing: A scoping review. *Cost Effectiveness and Resource Allocation: C/E*, 21:83, 2023.

[4] L. Ma and B. Sun. Machine learning and ai in marketing – connecting computing power to human insights. *International Journal of Research in Marketing*, 37(3):481–504, 2020.

[5] M. N. Omidvar, X. Li, and X. Yao. A review of population-based metaheuristics for large-scale black-box global optimization—part i. *IEEE Transactions on Evolutionary Computation*, 26(5):802–822, 2022.

[6] M. Abd Elaziz, A. Dahou, L. Abualigah, L. Yu, M. Alshinwan, A. M. Khasawneh, and S. Lu. Advanced metaheuristic optimization techniques in applications of deep neural networks: a review. *Neural Computing and Applications*, 33(21):14079–14099, 2021.

[7] G. Di Domenico, E. Andreis, A. Carlo Morelli, G. Merisio, V. Franzese, C. Giordano, A. Morselli, P. Panicucci, F. Ferrari, and F. Topputo. The erc-funded extrema project: Achieving self-driving interplanetary cubesats. In G. Fasano and J. D. Pintér, editors, *Modeling and Optimization in Space Engineering: New Concepts and Approaches*, pages 167–199. Springer International Publishing, 2023.

[8] W. Deng, R. Chen, B. He, Y. Liu, L. Yin, and J. Guo. A novel two-stage hybrid swarm intelligence optimization algorithm and application. *Soft Computing*, 16(10):1707–1722, 2012.

[9] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos. Feature selection for high-dimensional data. *Progress in Artificial Intelligence*, 5(2):65–75, 2016.

[10] L. C. Molina, L. Belanche, and A. Nebot. Feature selection algorithms: A survey and experimental evaluation. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 306–313, 2002.

[11] Tummala S. L. V. Ayyarao, N. S. S. Ramakrishna, R. M. Elavarasan, N. Polumahanthi, M. Rambabu, G. Saini, B. Khan, and B. Alatas. War strategy optimization algorithm: A new effective metaheuristic algorithm for global optimization. *IEEE Access*, 10:25073–25105, 2022.

[12] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4):28–39, 2006.

[13] F. Zhao, X. Hu, L. Wang, T. Xu, N. Zhu, and Jonrinaldi. A reinforcement learning-driven brain storm optimisation algorithm for multi-objective energy-efficient distributed assembly no-wait flow shop scheduling problem. *International Journal of Production Research*, 61(9):2854–2872, 2023.

[14] J. P. Papa, G. H. Rosa, A. N. de Souza, and L. C. S. Afonso. Feature selection through binary brain storm optimization. *Computers & Electrical Engineering*, 72:468–481, 2018.

[15] E. Osaba, E. Villar-Rodriguez, J. Del Ser, A. J. Nebro, D. Molina, A. LaTorre, P. N. Suganthan, C. A. Coello Coello, and F. Herrera. A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems. *Swarm and Evolutionary Computation*, 64:100888, 2021.

[16] A learning-based hybrid framework for dynamic balancing of exploration-exploitation: Combining regression analysis and metaheuristics. *Mathematics*, 9(16):1976, n.d.

[17] B. Abdollahzadeh and F. S. Gharehchopogh. A multi-objective optimization algorithm for feature selection problems. *Engineering with Computers*, 38(3):1845–1863, 2022.

[18] Amir Seyyedabbasi, Wadhah Zeyad Tareq Tareq, and Nebojsa Bacanin. An effective hybrid metaheuristic algorithm for solving global optimization algorithms. *Multimedia Tools and Applications*, pages 1–36, 2024.

[19] Q. Al-Tashi, S. J. Abdul Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian. Binary optimization using hybrid grey wolf optimization for feature selection. *IEEE Access*, 7:39496–39508, 2019.

[20] S. Arora and P. Anand. Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications*, 116:147–160, 2019.

[21] L.-Y. Chuang, S.-W. Tsai, and C.-H. Yang. Improved binary particle swarm optimization using catfish effect for feature selection. *Expert Systems with Applications*, 38(10):12699–12707, 2011.

[22] Raghav Prasad Parouha and Pooja Verma. Design and applications of an advanced hybrid meta-heuristic algorithm for optimization problems. *Artificial Intelligence Review*, 54(8):5931–6010, 2021.

[23] E.-S. M. El-Kenawy, M. M. Eid, M. Saber, and A. Ibrahim. Mbgwo-sfs: Modified binary grey wolf optimizer based on stochastic fractal search for feature selection. *IEEE Access*, 8:107635–107649, 2020.

[24] E. Emary, H. M. Zawbaa, and A. E. Hassanien. Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, 172:371–381, 2016.

[25] M. Ghaemi and M.-R. Feizi-Derakhshi. Feature selection using forest optimization algorithm. *Pattern Recognition*, 60:121–129, 2016.