



Modified Smell Agent Optimization Algorithm for Enhanced Exploration and Exploitation Capabilities

Yakubu Abdulrazak, Adejoh Idris Araga, Aliyu Sabo & Muhammad Isa Aliyu

^{1,2&3}Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna.

⁴Department of Electrical & Electronic Engineering, Nigerian Army University Biu Borno State-Nigeria.

ABSTRACT

Metaheuristic algorithms have gained prominence in solving diverse optimization problems owing to their straightforwardness, versatility, and derivative-free nature. The Smell Agent Optimization (SAO) algorithm is a recent metaheuristic inspired by the human sense of smell. It operates through three modes: sniffing, trailing, and random. The sniffing mode simulates how an agent perceives scent molecules, while the trailing mode imitates how an agent tracks a scent to find its source. The random mode introduces stochastic behaviour to help the algorithm escape local minima. Like other metaheuristics, SAO encounters issues such as getting trapped in local optima, balancing exploration and exploitation, and slow convergence due to its modes. The integration of chaotic maps has been demonstrated to boost the performance of such algorithms. Specifically, chaotic maps such as chebyshev map, circle map and piecewise map were incorporated into each mode of SAO, resulting in a modified version called Chaotic Smell Agent Optimization (cSAO). In the modification, Chebyshev modify the sniffing mode, circle map modifies the trailing mode and piecewise modify the randomness so as to enhance overall effectiveness and convergence speed. Experiments on thirty-seven benchmark functions showed that cSAO obtained the global best solution in 29 out of 37 which constituted 78.37% of the total benchmark functions. The SAO obtained the second-best results in 20 out of 37 which is 54.05% of the benchmark functions. This result shows an improvement of 24.31% performance of cSAO over SAO. The PSO obtained the global best solution in 17 out of 37 which is 45.94% of the benchmark function showing a 32.42% decrease as compare to the cSAO. The GWO also obtained the global best solution in 13 out of 37 which is 35.13% of the benchmark function showing a 43.23% decrease as compare to the cSAO. Also, in terms of ranking, the cSAO is ranked first with a final rank value of 1. The SAO, PSO and GWO are ranked second, third and fourth respectively with a final rank value of 2, 3 and 4 respectively. This superior performance of the cSAO over the three algorithms is expected because of the introduction of the chaotic maps in each of the mode to aid the search process.

ARTICLE INFO

Article History

Received: March, 2026

Received in revised form: May, 2026

Accepted: June, 2026

Published online: September, 2026

KEYWORDS

Chaotic Maps; Exploration; Exploitation;
Function Evaluation; Local Minima.

INTRODUCTION

In recent times, algorithms inspired by natural phenomena have garnered significant interest due to their success in addressing diverse

complex optimization challenges and their capacity to deliver competitive outcomes [1]. Optimization can be understood as a structured approach to finding the most appropriate solution

Corresponding author: Yakubu Abdulrazak

abdulrazakyakubu100@gmail.com

Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna.

© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved



for a specific problem, ensuring that predefined constraints are met and resources are utilized effectively. Deterministic optimization algorithms generally produce the same solution when identical initial conditions and parameters are provided.

In contrast, stochastic optimization techniques incorporate random processes into their search mechanisms, meaning that different runs may produce different solutions even when the initial conditions are unchanged [2]. Modern universal optimization techniques, computational intelligence, and soft-computing methods increasingly employ metaheuristic strategies derived from natural phenomena. Typically, optimization algorithms are classified into two main categories: deterministic methods and stochastic methods [3].

Earlier research has further divided stochastic optimization methods into heuristic and metaheuristic categories. Heuristic techniques tend to be tailored for specific types of optimization problems and are highly dependent on the problem context. Conversely, metaheuristic algorithms are generally problem-agnostic and can be applied across a wide variety of optimization challenges [4]. A key feature of metaheuristics is their capacity to maintain a balance between exploration (diversification) and exploitation (intensification).

Exploration allows the algorithm to search different regions of the solution space to find promising candidates, while exploitation refines the search around high-quality solutions to enhance their fitness. Achieving an appropriate balance between these two aspects increases the likelihood of finding the global optimum and prevents early convergence to local optima [5]. Many metaheuristic strategies are inspired by natural, biological, physical, or social phenomena, and have been successfully used to address numerous complex optimization problems [6].

Particle Swarm Optimization (PSO) is a prominent example of a metaheuristic algorithm inspired by natural phenomena, modelled after the coordinated movement and social interactions seen in bird flocks and fish schools [7]. Although PSO is effective, it has some limitations, notably

its relatively limited ability to perform local searches. This can lead the algorithm to become trapped in suboptimal solutions, diminishing the quality of convergence and introducing uncertainty into the results [8]. Several other algorithms inspired by biological behaviours have also been developed. For instance, the Firefly Algorithm (FA) mimics the flashing and attraction behaviours of fireflies [9], while Cat Swarm Optimization (CSO) is based on the hunting, tracking, and stalking actions of cats [10].

The Bat Algorithm (BA) models echolocation used by microbats for navigation and prey detection [11], and Ant Colony Optimization (ACO) draws inspiration from the foraging and route selection behaviours of ant colonies [12]. Similarly, the Dream Optimization Algorithm (DOA) is inspired by the human dreaming process, integrating features like memory retention, forgetting, and self-organization into the optimization process [13]. To enhance the balance between exploration and exploitation, DOA uses memory mechanisms, strategies for forgetting and replenishing information, and dream-sharing techniques. These features aim to improve the algorithm's ability to escape local optima while maintaining effective search capabilities.

The Smell Agent Optimization (SAO) algorithm is a recently devised metaheuristic inspired by the sense of smell and the trailing behavior observed in agents such as humans, dogs, sharks, insects, and bacteria to locate scent sources [7]. SAO has been effectively applied to a variety of problems, including path planning [8], capacitated vehicle routing [9], frequency stabilization in micro-grids [10], hybrid renewable energy systems [7], and coordination schemes for distribution networks [11]. However, like other metaheuristic algorithms, it encounters challenges such as maintaining a balance between exploration and exploitation, becoming trapped in local minima, and experiencing slow convergence.

LITERATURE REVIEW

The Smell Agent Optimization (SAO) algorithm is a recent metaheuristic inspired by the sense of smell and the trail-following behaviour exhibited by agents such as humans, dogs,

sharks, insects, and bacteria to locate scent sources [23]. SAO has been effectively applied to various problem domains, such as path planning [24], capacitated vehicle routing [25], frequency stabilization in micro-grids [26], hybrid renewable energy systems [23], and coordination schemes in power distribution networks [27]. Like other metaheuristics, SAO faces challenges related to balancing exploration and exploitation, susceptibility to becoming trapped in local minima, and slow convergence rates.

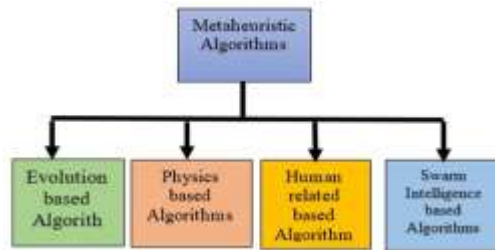


Fig. 1. Categorization of Metaheuristic Algorithms [21]

The rest of the paper is structured as follows. Section 2 discusses the core concepts of the Smell Agent Optimization and details how Chaotic maps mechanism has been integrated. Section 3 outlines the development of the chaotic Smell Agent Optimization (cSAO) and shares the results of its computational testing. Finally, Section 4 assesses the effectiveness of cSAO by comparing it with the standard MPA and various other optimization techniques through a series of experimental analyses.

Review of Metaheuristic Algorithm

This section focuses on the review of the metaheuristic algorithms such as Particle Swarm Optimization and the Marine Predator Optimization Algorithm as well as the levy flight utilized in this work.

Particle swarm optimization

Particle Swarm Optimization (PSO), a population-based algorithm inspired by social behaviours observed in nature such as fish schooling and bird flocking, was introduced by Kennedy and Eberhart in 1995 [28]. The method

begins by randomly generating a group of particles, mirroring the initialization step common in other evolutionary algorithms like genetic algorithms (GAs). Each particle keeps track of its own best experience, known as the personal best (pbest), along with the best position found by the entire swarm, called the global best (gbest).

Throughout the optimization process, particles iteratively update their velocities and positions based on both their personal findings and the swarm's collective knowledge. This dynamic process allows the particles to thoroughly explore the search space and gradually move toward optimal solutions. The updates for particle velocity and position are governed by two key equations [29], with each particle operating within a search space of D dimensions.

Position: $S_i = (s_{i1}, s_{i2}, \dots, s_{ik})$

Velocity: $v_i = (v_{i1}, v_{i2}, \dots, v_{ik})$

At each iteration, the velocity and position are updated as follows:

Velocity update Equation:

$$V_i^{k+1} = wV_i^k + c_1 * rand_1(P_{best} - S_i^k) + c_2 * rand_2(G_{best} - S_i^k) \quad (1)$$

$$S_i^{k+1} = S_i^k + V_i^{k+1} \quad (2)$$

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{iter_{max}} \right) \times iter \quad (3)$$

where,

V_i^k is the Velocity of agent i at kth iteration

V_i^{k+1} is the Velocity of agent i at (k+1)th iteration

w is the inertia weight

c_1 and c_2 are the weight factor (0 to 4)

S_i^k is the Current position of agent i at kth iteration

S_i^{k+1} is the Current position of agent i at (k+1)th iteration

$iter_{max}$ is the Maximum iteration number

iter is the Current iteration number

P_{best} is the personal best of agent i

G_{best} is the Global best of the group

w_{max} is the Initial value of the inertia weight

w_{min} is the Final value of the inertia weight

$rand_1$ and $rand_2$ are the random numbers usually chosen between [0,1].

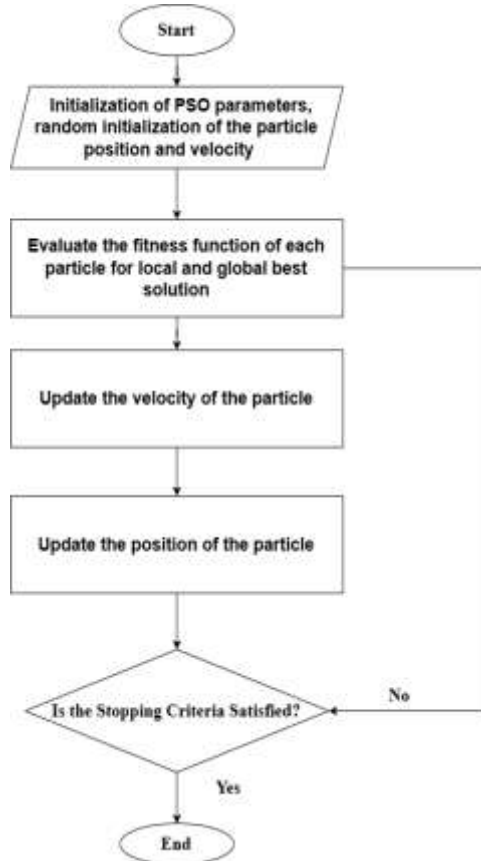


Figure 2: PSO Flowchart

Smell Agent Optimization Algorithm

A Smell Agent Optimization (SAO) algorithm is an optimization algorithm that uses olfactory phenomena and an agent's trailing behavior to identify a smell source. The algorithm works in three distinct modes: sniffing, trailing, and random mode. Each of these modes will be discussed as follows [23]:

Sniffing mode

The sniffing mode serves as the SAO's starting point. The smell molecules evaporate in the direction of the agent.

The agent sniffs the smell molecules and determines which molecule has the highest fitness. This is considered the location of an agent. In implementing the algorithm, the location of

these molecules is gotten by randomly generating the set of numbers as shown in equation (4) [23]:

$$X_i = \varphi(N, D) \quad (4)$$

Where φ is a random number generator which has a mean of zero and standard deviation of one. N is the population of the smell molecules number of the smell molecules considered as the decision variable. D is the dimension or search region of the algorithm.

The individual members of the population are assigned a vector position using the equation

$$X_i^{(t)} = [X_{N,1}^t, X_{N,2}^t, \dots, X_{N,D}^t] \quad (5)$$

This vector position enables an agent to determine the region with the highest concentration in the solution search space. Also, since the smell molecules are moving in the direction of the agent, they move with a uniform velocity. Therefore, the velocity of the smell molecule can also be represented in a vector as :

$$V_i^{(t)} = [V_{N,1}^t, V_{N,2}^t, \dots, V_{N,D}^t] \quad (6)$$

The evaporation of smell molecule is given by

$$X_i^{(t+1)} = X_i^t + V_i^t \quad (7)$$

As the agent moves in the solution search space the velocity and position of smell molecules are updated with the equations 8 and 9 respectively

$$V_i^{(t+1)} = V_i^t + r_1 \times \left(\frac{3kT}{m} \right) \quad (8)$$

$$X_i^{(t+1)} = X_i^t + V_i^{(t+1)} \quad (9)$$

where $V_i^{(t+1)}$ is the updated velocity, V_i^t is the previous velocity, k is the Boltzmann's constant, T is the absolute temperature of the smell

molecules, r_1 is a constant which stochastically penalize the velocity update.

Trailing mode

In the trailing mode, the agent follows the molecule with a higher concentration because it is believed that as the agent approaches the smell source, the concentration/intensity of the

$$X_i^{(t+1)} = X_i^t + r_2 \times ofc \times (X_{Agent}^t - X_i^t) - r_3 \times ofc \times (X_{worst}^t - X_i^t) \quad (10)$$

Where *ofc* is the olfaction capacity of an agent, r_2 and r_3 are random numbers generated at different intervals

Random Mode

The random mode on the other hand occurs when an agent is confused or trapped in the local minimum during trailing. This mode helps the agent to assume any random step in the

smell molecules increases. Thus, the concentration of the smell molecule is critical during trailing. Furthermore, an agent's olfactory capacity should be strong to aid the process. During trailing, the agent observes the molecules with the best and worst fitness, allowing the agent to follow an optimal path while trailing. The mathematical equation governing the trailing mode is as shown in equation 10.

search space to continue trailing or the agent goes back and starts the whole process again [23].

The random step is represented by the mathematical equation 11.

$$X_i^{(t+1)} = X_i^t + r_4 \times \theta \quad (12)$$

where θ is the step movement, r_4 is the random number which stochastically penalises the step movement. The flow of the algorithms is illustrated with the flow chart as shown in Figure 2.2.

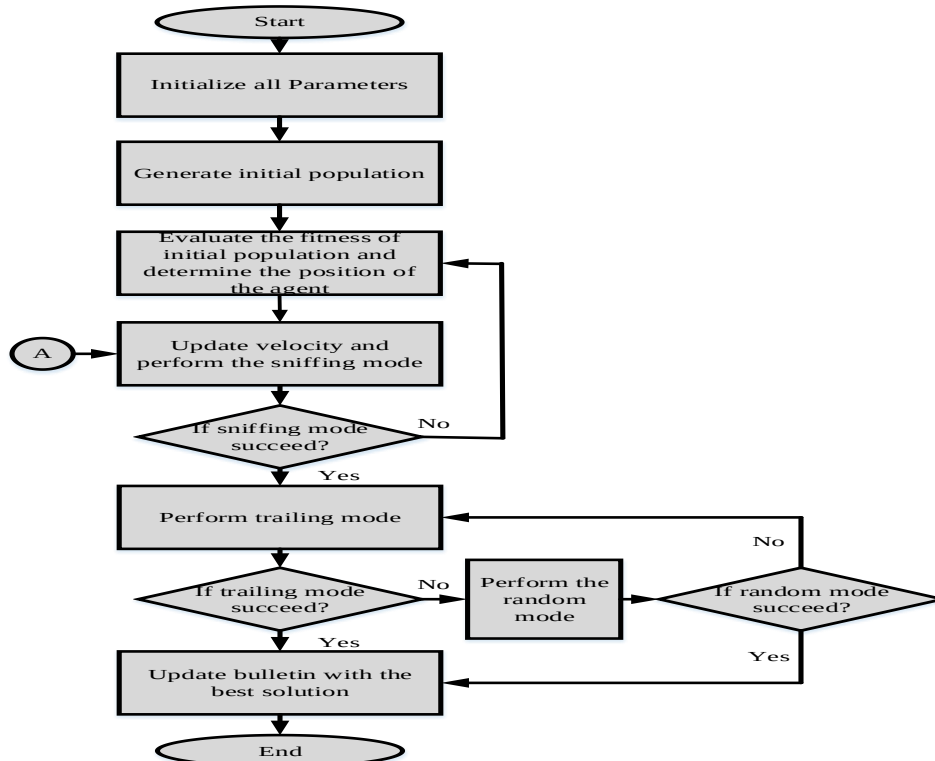


Figure 2.2: Flow Chart of SAO

Corresponding author: Yakubu Abdulrazak

abdulrazakyakubu100@gmail.com

Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna.

© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved



Development of a Chaotic Smell Agent Optimization Algorithm

The modification on the original SAO is targeted at improving the exploitation capability of the algorithm while not undermining the exploration ability as well. As reported in the problem statement, original SAO obtain an optimal result at the expense of time. To improve its convergence ability, chaotic maps such as piecewise map, chebyshev map and circle map is incorporated in the sniffing, trailing and random mode to guide the searching process. The detail mathematical expression of each of the modified mode with chaotic maps is as show in the next section.

Sniffing mode of cSAO

From the original SAO the sniffing mode model the evaporation of the smell molecules in the direction of an agent. An agent could not account for all the smell molecule been evaporated in its direction. Therefore, this led to a high level of exploration ability in this mode. The equation governing the sniffing mode is represented Section 2. The parameter r_1 is an important parameter that penalize the sniffing mode. In the original SAO this is obtain using standard probability distribution. In the chaotic SAO this can be generated using the Chebyshev chaotic map described by the mathematical expression in equation (10) from section 2.6.1. Where x_{n+1} is the chebyshev chaotic number to be generated, x_n is the current position of the map. From equation (11), if $n=0$, the current position becomes the starting position known represented as x_0 and the chebyshev chaotic map becomes x_1 which simplify the equation as:

$$X_{n+1} = \cos(K \cdot \cos^{-1} X_n) \quad (13)$$

This equation (14) is transformed into a function as $n = 0$

$$x_1 = \cos(K \cdot \cos^{-1} x_0) \quad (14)$$

Therefore, the chaotic sniffing mode can be expressed as

$$X_i^{(t+1)} = X_i^t + V_i^{(t+1)} \quad (15)$$

$$V_i^{(t+1)} = V_i^t + r_1 \times (3kT)/M \quad (16)$$

From equation (14) substitute equation (15)

$$X_i^{(t+1)} = X_i^t + V_i^t + r_1 \times (3kT)/M \quad (17)$$

$$r_1 = x_1$$

$$X_i^{(t+1)} = X_i^t + V_i^t + (\cos(K \cdot \cos^{-1} x_0) \times (3kT)/M,$$

$$\text{Therefore } X_i^{(t+1)} = X_i^t + V_i^t + \text{Chebyshev}(r) \times (3kT)/M \quad (18)$$

Where Chebyshev (r) is a random number generated by a chebyshev chaotic map and r is any arbitrary random number in the range of 0 and 1. The introduction of the chebyshev chaotic map is targeted at enhancing the exploitation ability as well as the convergence time of the algorithm.

Trailing mode of CSAO

The trailing mode of the original SAO model the intuitive behavior of an agent trailing the molecules for optimal solution. The mathematical expression is described using equation 2.28 in Section 2.6.2. The random number r_2 and r_3 are generated using circle chaotic map represented as c_1 and c_2 described by the expression shown in equation 2.32 in Section 2.6.2 as

$$x_{n+1} = x_n + B - \frac{K}{2\pi} \times \sin(2\pi x_n) \text{ Mod } 1 \quad (19)$$

When $n = 1$ the expression becomes

$$x_2 = x_1 + B - \frac{K}{2\pi} \times \sin(2\pi x_1) \text{ mod } 1 \quad (20)$$

x_1 is the current chaotic value, x_2 is the next chaotic value generated using random number in the range of 0 and 1 while K is a nonlinear parameter, B is the frequency parameter, while $K = 0.5$ to 6, $B = 0.2$ to 5 for a chaotic system.

Therefore, the chaotic trailing mode is express as:

$$X_i^{(t+1)} = X_i^t + c_1 \times \text{ofc} \times (X_{\text{agent}}^t - X_i^t) - c_2 \times \text{ofc} \times (X_{\text{worst}}^t - X_i^t) \quad (21)$$

Assuming $X_2 = C_2$ therefore the trailing equation upon introduction of the circle map

$$X_i^{(t+1)} = X_i^t + c_1 \times \text{ofc} \times (X_{\text{agent}}^t - X_i^t) - (x_1 B - \frac{K}{2\pi} \times \sin(2\pi x_1) \text{ mod } 1) \times \text{ofc} \times (X_{\text{worst}}^t - X_i^t)$$

$$X_i^{(t+1)} = X_i^t + c_1 \times \text{ofc} \times (X_{\text{agent}}^t - X_i^t) - \text{circle map}(c) \times \text{ofc} \times (X_{\text{worst}}^t - X_i^t) \quad (22)$$

This modification is aim at increasing the searching speed of the mode and subsequently converges to an optimal solution in less time.

Random mode of CSAO

The random mode equation in section 2. The parameter r_4 is also generated chaotically using piecewise map described by equation 23. Therefore, the chaotic random mode equation is expressed as

$$X_i^{(t+1)} = X_i^t + r_4 \times \theta \quad (23)$$

For $n=3$, substitute n into this equation of piecewise chaotic map

$$X_{n+1} = \begin{cases} \frac{Xn}{P} & 0 \leq Xn \leq P \\ \frac{Xn-P}{0.5-P} & P \leq Xn \leq 0.5 \\ \frac{1-P-Xn}{0.5-P} & 0.5 \leq Xn \leq 1-P \\ \frac{1-Xn}{P} & 1-P \leq Xn \leq 1 \end{cases} \quad (24)$$

Substituting $n=3$

$$X_4 = \begin{cases} \frac{X_3}{P} & 0 \leq X_3 \leq P \\ \frac{X_3-P}{0.5-P} & P \leq X_3 \leq 0.5 \\ \frac{1-P-X_3}{0.5-P} & 0.5 \leq X_3 \leq 1-P \\ \frac{1-X_3}{P} & 1-P \leq X_3 \leq 1 \end{cases} \quad (25)$$

P is the control parameter with value range of $0 < P < 0.5$, P is approximately $=0.4$ Where θ is the step movement, r_4 is the random number which stochastically penalises the step

$$X_i^{(t+1)} = X_i^t + \text{piecewise map}(r) \times \theta, X_4 = r_4$$

$$X_i^{(t+1)} = X_i^t + \begin{cases} \frac{X_3}{P} & 0 \leq X_3 \leq P \\ \frac{X_3-P}{0.5-P} & P \leq X_3 \leq 0.5 \\ \frac{1-P-X_3}{0.5-P} & 0.5 \leq X_3 \leq 1-P \\ \frac{1-X_3}{P} & 1-P \leq X_3 \leq 1 \end{cases} \times \theta \quad (26)$$

Evaluation of cSAO on Benchmark Functions

The performance of an optimization algorithm cannot be accurately assessed by the specific problems it addresses, especially if those problems are highly specialized and lack diverse characteristics. To thoroughly evaluate the proposed algorithm's effectiveness, a selection of seven benchmark functions will be used, chosen according to the attributes outlined in Table 1. The complexity of each of these functions is represented through a corresponding metric function

Table 1: Benchmark Function

Fn	Names	Description
F ₁	Adjiman	$f(x_1, x_2) = \cos(x_1) \sin(x_2) - \frac{x_1}{x_2^2 + 1}$
F ₂	Bohachevsky1	$f(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$
F ₃	Michalewicz	$f(x) = -\sum_{i=1}^d \sin(x_i) \sin^{2m} \left(\frac{ix_i^2}{\pi} \right)$
F ₄	Schaffer	$f(x) = \sum_{i=1}^{30} (x_i^2 + x_{i+1}^2)^{0.5} \left\{ \left[\sin 50 (x_i^2 + x_{i+1}^2)^{0.1} \right]^2 \right\}$
F ₅	Styblinski's Tang	$f(x) = \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$
F ₆	Box-Betts	$f(x_1, x_2) = \sum_{i=1}^k (e^{-0.1(i+1)x_1} - e^{-0.1(i+1)x_2} - [(e^{-0.1(i+1)}) - e^{-(i+1)x_3}])^2$



F _n	Names	Description
F ₇	Michalewicz 5	$f(x) = -\sum_{i=1}^d \sin(x_i) \sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$
F ₈	Miele Cantrell	$f(X) = (e^{-x_1} - e^{-x_2})^4 + 100(x_2 - x_3)^6 + (\tan(x_3 - x_4))^4 + x_1^8$
F ₉	Step	$f(x) = \sum_{i=1}^n (x_i + 0.5)^2$
F ₁₀	Michalewicz 10	$f(x_1, x_2) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$
F ₁₁	Brown	$f(X) = \sum_{i=1}^{n-1} (x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)}$
F ₁₂	Ellipsoid	$f(x) = \sum_{i=1}^n i^2 x_i^2$
F ₁₃	Quartic	$f(X) = \sum_{i=1}^D ix_i^4 + \text{random}[0,1]$
F ₁₄	Rosenbrock	$f(x) = \sum_{i=1}^{n-1} ((x_i - 1)^2 + 100(x_{i+1} - x_i^2)^2)$
F ₁₅	Salomon	$f(X) = 1 - \cos(2\pi \sqrt{\sum_{i=1}^n x_i^2}) + 0.1 \sqrt{\sum_{i=1}^n x_i^2}$

RESULTS AND DISCUSSION

The cSAO was successfully developed using the Chebyshev map, Circle Map and Piece wise map. The flow chart in Figure 3.7 illustrates the flow of the developed algorithm. All the parameters necessary for successful implementation of the algorithm were initialized. Then the initial population of the smell molecules was generated. The fitness of the smell molecules was evaluated and the molecule with the best fitness is selected as the agent. Then the Chebyshev chaotic random number was generated and the chaotic sniffing mode was executed. The position of the agent is updated and the molecule with the worst fitness is determined. Then a Circle chaotic map was used to generate

random number to aid the search process in the chaotic trailing mode.

The fitness of the trailing mode is evaluated and the position of the agent is updated and the best solution obtained as displaced at the output. If an agent fails to obtain the best solution during trailing a chaotic piece wise map is used to generate chaotic random number that will launch the random mode. If the random mode obtained the best solution, it is displaced at the output terminal and if it fails, it returns back to start the process again. The introduction of a chaotic map prior to the start of each mode is targeted at enhancing the search process to obtain better result and also converges faster.

Corresponding author: Yakubu Abdulrazak

abdulrazakyakubu100@gmail.com

Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna.

© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved



Performance of the Algorithms on Benchmark Functions

Tables 4.1 show the experimental result for the benchmark function. The result obtained for the 2D, 3D, 4D, 5D, and 10D benchmark functions are shown in Table 4.1. To achieve the best outcome, the developed CSAO was assessed for each benchmark function in conjunction with other metaheuristic algorithms like the original SAO, PSO, and GWO. These algorithms were run fifty (50) times and each run execute one-thousand (1000) iterations. The best, worst, average, standard deviation, and rank of each algorithm

were determined after fifty runs. The algorithms' performance is ranked based on the best value obtained from the benchmark function. For instance, if p algorithms all returned the same best value for the benchmark function, they would all rank the same, and the next algorithm would rank $p+1$. The average rank on the benchmark functions was computed at the end of each table. The algorithms with the best performance were then identified by calculating the average rank, which also served to determine the final rank of the algorithms.

Table 4.3 Selected best results on 2D, 3D, 4D and 10D benchmark functions

Fn	Performance metrics	CSAO	SAO	PSO	GWO	Global best
F1	Mean	-2.0218	-2.0218	-2.0209	-2.098	-2.0218
	Best	-2.0218	-2.0218	-2.0218	-2.0218	
	Std	0.000	0.000	1.20E-03	-3.451E-02	
	Worst	-2.0218	-2.0218	-2.0202	-1.590	
	Rank	1	1	3	4	
F2	Mean	2.456E-04	3.226E-04	0.0000	1.17E-01	0
	Best	4.745E-06	1.688E-05	0.0000	9.314E-04	
	Std	2.364E-04	3.180E-04	0.0000	2.97E-01	
	Worst	8.205E-04	1.371E-03	0.0000	8.78E-01	
	Rank	2	3	1	4	
F3	Mean	-106.776	-106.775	-106.787	-103.674	-106.765
	Best	-106.765	-106.765	-106.787	-106.787	
	Std	9.32E-03	1.50E-02	7.095E-14	7.081	
	Worst	-106.787	-106.787	-106.787	-87.866	
	Rank	1	2	3	4	
F4	Mean	0.0000	0.0000	0.0000	0.0000	0
	Best	0.0000	0.0000	0.0000	0.0000	
	Std	0.0000	0.0000	0.0000	0.0000	
	Worst	0.0000	0.0000	0.0000	0.0000	
	Rank	1	1	1	1	
F5	Mean	2.120E-03	0.0000	2.073E-03	7.441E-08	0
	Best	7.000E-05	0.0000	1.604E-05	1.739E-09	
	Std	1.897E-03	0.0000	2.365E-03	6.714E-08	
	Worst	6.321E-03	0.0000	1.002E-02	2.731E-07	
	Rank	4	1	3	2	
F6	Mean	0.3982	0.3980	0.39788	0.4006	0.3979
	Best	0.3979	0.3979	0.3978	0.3979	
	Std	3.873E-04	2.057E-04	0.000	234E-03	
	Worst	0.3998	0.3987	0.39788	0.4051	
	Rank	2	1	4	3	
F7	Mean	-0.1773	-0.1779	-0.1983	-0.2018	

Corresponding author: Yakubu Abdulrazak

abdulrazakyakubu100@gmail.com

Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna.

© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved



Fn	Performance metrics	CSAO	SAO	PSO	GWO	Global best
F8	Best	-0.1799	-0.1799	-0.2779	-0.2755	-0.1799
	Std	2.12E-04	1.60E-04	2.32E-02	5.03E-02	
	Worst	-0.1760	-0.1769	-0.1892	-0.2009	
	Rank	2	1	3	4	
	Mean	1.383E-87	1.383E-87	1.383E-87	4.321E-48	0
F9	Best	1.383E-87	1.383E-87	1.383E-87	1.383E-87	
	Std	6.811E-103	6.811E-103	6.811E-103	1.23E-48	
	Worst	1.383E-87	1.383E-87	1.383E-87	2.121E-45	
	Rank	1	1	1	4	
	Mean	1.837	1.741	2.0974	2.0975	0
F10	Best	1.832	1.724	2.0974	2.0975	
	Std	3.765E-03	9.73E-03	1.806E-15	2.6153E-14	
	Worst	1.845	1.765	2.0974	2.0975	
	Rank	2	1	3	4	
	Mean	-1.0316	-1.0316	-1.0316	-1.03162	-1.0316
F11	Best	-1.0316	-1.0316	-1.0316	-1.03162	
	Std	0.000	0.000	0.000	1.529E-08	
	Worst	-1.0316	-1.0316	-1.0316	-1.03162	
	Rank	1	1	1	4	
	Mean	-42.940	-42.907	-42.825	-35.173	-42.944
F12	Best	-42.944	-42.944	-42.944	-42.944	
	Std	3.472E-02	3.747E-02	2.014E-01	7.305	
	Worst	-42.788	-42.811	-42.497	-20.772	
	Rank	1	2	3	4	
	Mean	-24776	-24773	-24776	24775	-24777
F13	Best	-24777	-24777	-24776	-24776	
	Std	1.8E-15	2.71E-14	3.70E-12	2.472E-01	
	Worst	-24706	-24700	-24776	24736	
	Rank	1	2	3	4	
	Mean	-1.000	-1.000	-1.000	-0.200	-1
F14	Best	-1.000	-1.000	-1.000	-0.9991	
	Std	0.000	0.000	0.000	4.06E-01	
	Worst	-1.000	-1.000	-1.000	-8.110E-05	
	Rank	1	1	1	3	
	Mean	0.0000	0.0000	0.0000	0.0000	0
F15	Best	0.0000	0.0000	0.0000	0.0000	
	Std	0.0000	0.0000	0.0000	0.0000	
	Worst	0.0000	0.0000	0.0000	0.0000	
	Rank	1	1	1	1	
	Mean	-1.9132	-1.9131	-0.2660	-1.9130	-1.9133
F16	Best	-1.9132	-1.9132	-1.9132	-1.9132	
	Std	1.469E-05	1.534E-05	4.4999	2.345E-05	
	Worst	-1.9132	-1.9130	22.3700	-1.9129	
	Rank	1	2	4	3	
	Mean	-1.9999	-1.9999	-2.000	-1.9965	-1.8013
	Best	-1.9999	-1.9998	-2.000	-1.9999	

Corresponding author: Yakubu Abdulrazak

abdulrazakyakubu100@gmail.com

Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna.

© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved



F _n	Performance metrics	CSAO	SAO	PSO	GWO	Global best
F17	Std	2.309E-05	1.773E-05	0.000	8.480E-03	
	Worst	-1.9999	-1.9999	-2.000	-1.9598	
	Rank	2	1	4	3	
	Mean	-3873.63	-3873.62	-3873.72	-3873.72	-
	Best	-3873.72	-3873.72	-3873.72	-3873.72	3873.72
	Std	1.02E-02	0.101	1.387E-12	2.884E-06	
F18	Worst	-3873.59	-3873.39	-3873.72	-3873.72	
	Rank	3	4	1	2	
	Mean	0.0000	0.0000	5.320E-30	0.0000	0
	Best	0.0000	0.0000	1.397E-30	0.0000	
	Std	0.0000	0.0000	8.730E-30	0.0000	
	Worst	0.0000	0.0000	2.719E-29	0.0000	
F19	Rank	1	1	4	1	
	Mean	-78.329	-78.329	-78.332	-76.457	-78.332
	Best	-78.332	-78.332	-78.332	-78.332	
	Std	2.93E-03	2.27E-03	1.445E-14	4.862	
	Worst	-78.320	-78.323	-78.332	-64.260	
	Rank	3	2	1	4	
F20	Mean	0.0000	0.0000	0.0000	0.0000	0
	Best	0.0000	0.0000	0.0000	0.0000	
	Std	0.0000	0.0000	0.0000	0.0000	
	Worst	0.0000	0.0000	0.0000	0.0000	
	Rank	1	1	1	1	
	mean	0.0000	0.0000	0.0000	0.9069	0
F21	Best	0.0000	0.0000	0.0000	2.257E-06	
	Std	0.0000	0.0000	0.0000	1.6719	
	Worst	0.0000	0.0000	0.0000	3.886	
	Rank	1	1	1	4	
	mean	7.558E-08	1.237E-07	1.762E-022	0.0000	0
	Best	1.693E-10	6.631E-10	3.256E-023	0.0000	
F22	Std	1.099E-07	1.360E-07	6.122E-022	0.0000	
	Worst	4.606E-07	5.559E-07	2.456E-020	0.0000	
	Rank	3	4	2	1	
	mean	-4.999	-4.998	-5.000	-4.9812	-4.6877
	Best	-4.999	-4.999	-5.000	-4.9993	
	Std	5.35E-05	6.21E-05	0.000	2.075E-02	
F23	Worst	-4.999	-5.000	-5.000	-4.8973	
	Rank	1	2	4	3	
	mean	0.0000	3.487E-17	3.321E-16	5.432E-04	0
	Best	0.0000	2.132E-19	9.421E-17	2.991E-05	
	Std	0.0000	1.980E-14	3.423E-15	3.546E-04	
	Worst	0.0000	3.212E-12	5.342E-15	6.452E-04	
F24	Rank	1	2	3	4	
	mean	0.0000	0.0000	0.0000	0.0000	0
	Best	0.0000	0.0000	0.0000	0.0000	
F25	Std	0.0000	0.0000	0.0000	0.0000	

Corresponding author: Yakubu Abdulrazak

abdulrazakyakubu100@gmail.com

Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna.

© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved



Fn	Performance metrics	CSAO	SAO	PSO	GWO	Global best
F26	Worst	0.0000	0.0000	0.0000	0.0000	
	Rank	1	1	1	1	
	mean	-9.9999	-9.9999	-10.000	-9.9330	-9.6602
	Best	-9.9996	-9.9996	-10.000	-9.9983	
	Std	1.00E-05	8.886E-05	0.000	1.91E-02	
F27	Worst	-9.9999	-9.9999	-10.000	-9.9330	
	Rank	1	2	4	3	
	mean	-186.73	-186.68	-186.69	-152.02	-186.73
	Best	-186.73	-186.73	-186.73	-186.73	
	Std	2.41E-14	3.87E-02	3.25E-02	39.661	
F28	Worst	-186.73	-186.60	-186.59	-55.736	
	Rank	1	3	2	4	
	mean	8.8817E-16	8.8817E-16	8.8817E-16	8.8817E-16	0
	Best	8.8817E-16	0.0000	0.0000	8.8817E-16	
	Std	0.0000	1	1	0.0000	
F29	Worst	8.8817E-16			8.8817E-16	
	Rank	1			1	
	mean	0.0000	0.0000	0.0000	0.0000	0
	Best	0.0000	0.0000	0.0000	0.0000	
	Std	0.0000	0.0000	0.0000	0.0000	
F30	Worst	0.0000	0.0000	0.0000	0.0000	
	Rank	1	1	1	1	
	mean	0.0000	0.0000	0.0000	0.0000	0
	Best	0.0000	0.0000	0.0000	0.0000	
	Std	0.0000	0.0000	0.0000	0.0000	
F31	Worst	0.0000	0.0000	0.0000	0.0000	
	Rank	1	1	1	1	
	mean	0.0000	0.0000	0.0000	0.0000	0
	Best	0.0000	0.0000	0.0000	0.0000	
	Std	0.0000	0.0000	0.0000	0.0000	
F32	Worst	0.0000	0.0000	0.0000	0.0000	
	Rank	1	1	1	1	
	mean	5.282E-51	1.092E-25	2.000	6.798	0
	Best	1.417E-54	9.606E-29	2.000	2.8182	
	Std	1.061E-51	1.687E-25	1.722E-14	4.4429	
F33	Worst	4.018E-50	6.978E-25	2.000	24.5729	
	Rank	1	2	3	4	
	mean	1.375E-04	5.021E-04	1.029E-04	5.227E-04	0
	Best	1.377E-06	1.625E-06	6.652E-06	3.3186E-06	
	Std	1.140E-04	4.188E-04	2.988E-04	5.377E-04	
F34	Worst	4.393E-03	1.575E-03	8.369E-05	2.884E-03	
	Rank	1	2	4	3	
	mean	0.0000	0.0000	0.0000	0.0000	0

Corresponding author: Yakubu Abdulrazak

abdulrazakyakubu100@gmail.com

Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna.

© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved



F _n	Performance metrics	CSAO	SAO	PSO	GWO	Global best
F35	Best	0.0000	0.0000	0.0000	0.0000	
	Std	0.0000	0.0000	0.0000	0.0000	
	Worst	0.0000	0.0000	0.0000	0.0000	
	Rank	1	1	1	1	
	mean	0.0000	7.877E-03	7497E-03	3.0014E-02	0
F36	Best	0.0000	6.506E-04	3.815E-05	3.8732E-05	
	Std	0.0000	7877E-03	7.857E-03	1.041E-01	
	Worst	0.0000	2.636E-02	3.745E-02	5813E-01	
	Rank	1	4	2	3	
	mean	0.0000	0.0000	7.989E-02	0.0000	0
F37	Best	0.0000	0.0000	0.0000	0.0000	
	Std	0.0000	0.0000	4.063E-02	0.0000	
	Worst	0.0000	0.0000	9.987E-02	0.0000	
	Rank	1	1	4	1	
	mean	0.0000	0.0000	0.0000	0.0000	0
Global best		29	20	17	13	
Average		1.276	1.850	2.176	2.846	
Ranking						
Final Ranking		1	2	3	4	

From the result shown in Table 4.3, it was observed that the cSAO obtained the global best solution in 29 out of 37 which constituted 78.37% of the total benchmark functions. The SAO obtained the second-best results in 20 out of 37 which is 54.05% of the benchmark functions. This result shows an improvement of 24.31% performance of cSAO over SAO. This percentage improvement is calculated using equation 3.169 in section 3.9.1 The PSO obtained the global best solution in 17 out of 37 which is 45.94% of the benchmark function showing a 32.42% decrease as compare to the cSAO. The GWO also obtained the global best solution in 13 out of 37 which is 35.13% of the benchmark function showing a 43.23% decrease as compare to the cSAO. Also, in terms of ranking, the cSAO is ranked first with a final rank value of 1. The SAO, PSO and GWO are ranked second, third and fourth respectively with a final rank value of 2, 3 and 4 respectively. This superior performance of the cSAO over the three algorithms is expected because of the introduction

of the chaotic maps in each of the mode to aid the search process.

Convergence Analysis

As a result of the unique characteristics of metaheuristic algorithms, it is usually unfair to analyze the convergence and execution speed base on iteration. This is because iteration does not provide an even base for comparison in the sense that some algorithms can evaluate the objective function two or three times in a given iteration. For example, in each iteration, the objective function can be evaluated four times by both the original SAO and the CSAO. Every iteration, the PSO and GWO are each capable of evaluating the objective function twice and five times, respectively.

Function evaluation will be taken into consideration to guarantee an even basis for comparison. The iteration number also known as the terminating criteria is used to compute the function evaluation. For the purposes of this work,

Corresponding author: Yakubu Abdulrazak

abdulrazakyakubu100@gmail.com

Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna.

© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved

the maximum number of iterations is 1000. The iteration table, which shows the number of iterations required for an algorithm to converge, was obtained first. If an algorithm obtained the global solution at the first iteration, the iteration number is set as 1 and if an algorithm fails to obtain the global solution for the maximum number of iteration then the iteration number is set as 1000. A total of twelve (12) benchmark functions were selected randomly to investigate the convergence of the CSAO.

The way the solution approaches the global optimum influenced the selection of these functions. For instance, Figure 4.8 to Figure 4.19 presented the convergence plots for CSAO and other algorithms.

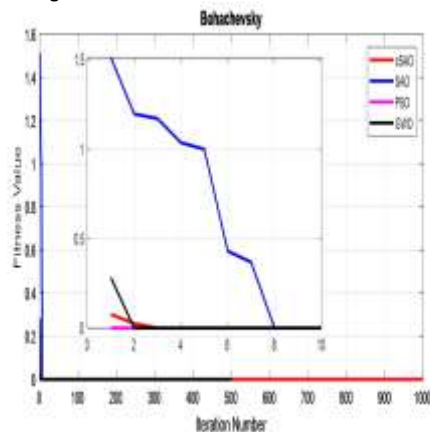


Figure 4.8: Convergence Plot for Bohachevsky Function

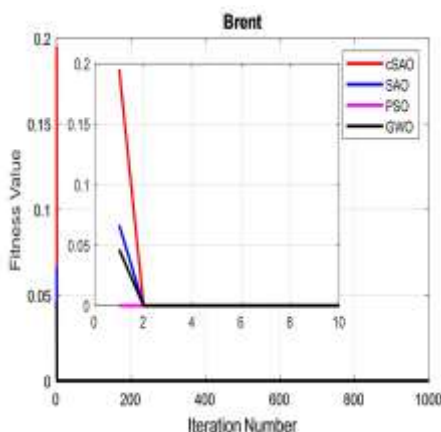


Figure 4.9: Convergence Plot for Brent Function

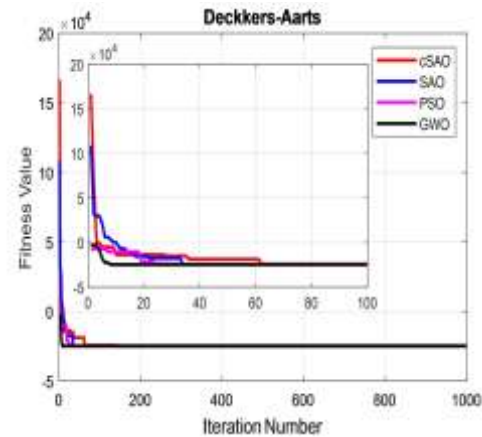


Figure 4.10: Convergence Plot for Deckkers-Aarts Function

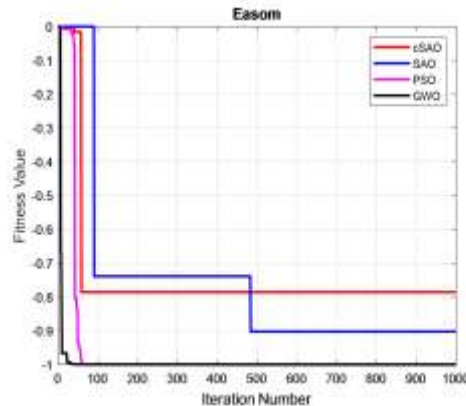


Figure 4.11: Convergence Plot for Easom Function

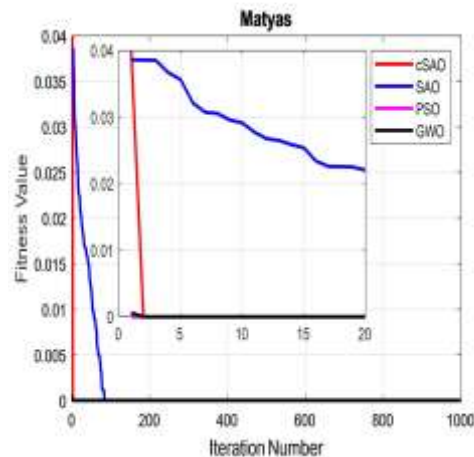


Figure 4.12: Convergence Plot for Matyas Function

Corresponding author: Yakubu Abdulrazak

abdulrazakyakubu100@gmail.com

Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna.

© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved

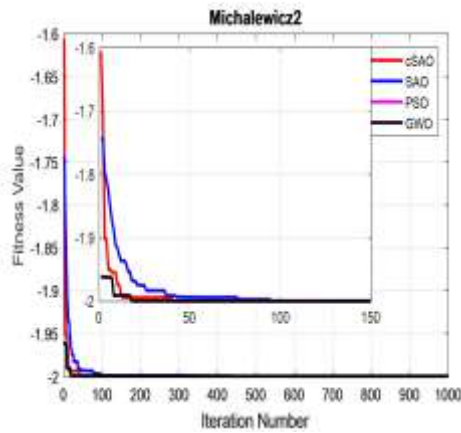


Figure 4.13: Convergence Plot for Michalewicz2 Function

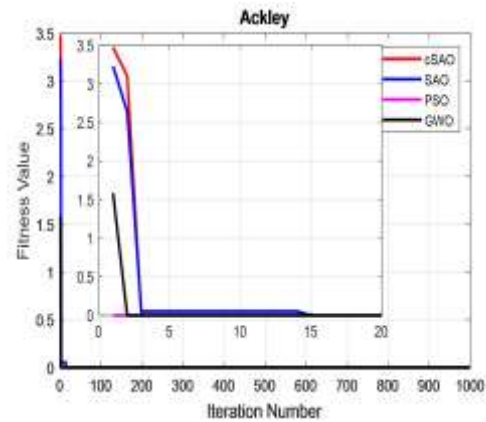


Figure 4.16: Convergence Plot for Ackley Function

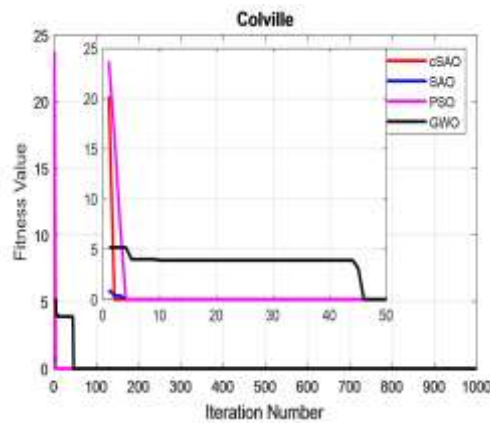


Figure 4.14: Convergence Plot for Colville Function

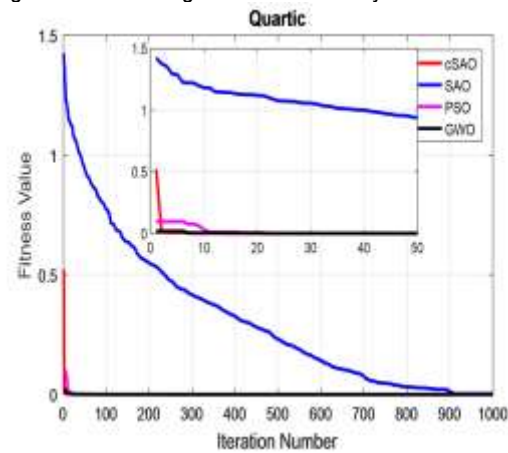


Figure 4.17: Convergence Plot for Quartic Function

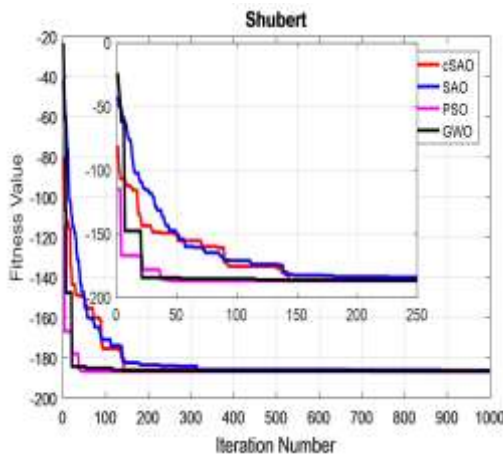


Figure 4.15: Convergence Plot for Shubert Function

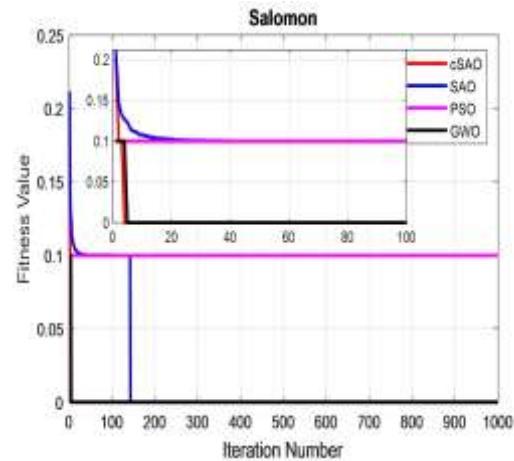


Figure 4.18: Convergence Plot for Salomon Function

Corresponding author: Yakubu Abdulrazak

abdulrazakyakubu100@gmail.com

Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna.

© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved

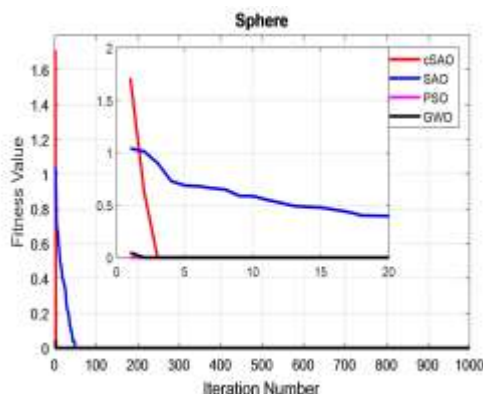


Figure 4.19: Convergence Plot for Sphere Function

Table 4.4 Selected best results on 2D-10D benchmark functions for convergence

FUNCTION	cSAO	SAO	PSO	GWO
F4	300	600	180	250
F8	220	220	180	220
F12	280	280	240	100
F13	60	90	80	70
F14	180	900	150	180
F16	150	450	900	180
F19	140	160	160	740
F21	540	650	250	180
F26	200	200	150	160
F27	220	900	340	300
F28	130	300	150	150
F32	250	900	190	220
$\sum (F4 \dots F32)$	2670	5650	2970	2750
Accelerated Rate	Reference value $AR = \frac{FE_i}{FE_f}, AR_{csao} = 1$			
		$\frac{5650}{2670} = 2.12$	$\frac{2970}{2670} = 1.1$	$\frac{2750}{2670} = 1.03$

From Table 4.4, it was observed that CSAO convergences faster when compare to the original SAO. This statement is justified by Accelerated Rate [98] value obtained in Table 4.4 which showed that $AR > 1$ for the average of all SAO functions that is $AR = 2.116$. The CSAO converges faster than PSO according to the value in table 4.4 which shows that PSO $AR = 1.12$ and CSAO converges faster than the GWO with AR value = 1.030.

CONCLUSION

This paper presented a new metaheuristic algorithm known as chaotic smell agent optimization (cSAO) that uses the concept of chaotic map in each of the modes of SAO. The cSAO operates in three distinct modes known as chaotic sniffing, trailing and random mode just like the original SAO. The chaotic maps introduced in each of the modes aid the search process and improve the performance of the algorithm. The performance of cSAO was evaluated on seventeen benchmark functions with different properties. Statistical results obtained showed that



the cSAO outperformed the SAO, PSO and GWO with a final rank value of 2, 3 and 4 respectively.

The convergence analysis also showed that cSAO obtained the global best solution in 29 out of 37 which constituted 78.37% of the total benchmark functions. The SAO obtained the second-best results in 20 out of 37 which is 54.05% of the benchmark functions. This result shows an improvement of 24.31% performance of cSAO over SAO. This percentage improvement is calculated using equation 3.169 in section 3.9.1 The PSO obtained the global best solution in 17 out of 37 which is 45.94% of the benchmark function showing a 32.42% decrease as compare to the cSAO. The GWO also obtained the global best solution in 13 out of 37 which is 35.13% of the benchmark function showing a 43.23% decrease as compare to the cSAO. Also in terms of ranking, the cSAO is ranked first with a final rank value of 1

REFERENCES

- [1] Yang, X.-S. "Nature-inspired metaheuristic algorithms: Success and new challenges". 2012. p1211.6658.
- [2] Yılmaz, S., & Küçüksille, E. U. "A new modification approach on bat algorithm for solving optimization problems". *Applied Soft Computing*, 2015 p28, 259-275.
- [3] Yang, X.-S. "Firefly algorithm, stochastic test functions and design optimization". 2010a. *International Journal of Bio-Inspired Computation*, 2(2), 78-84.
- [4] Shehab, M., Khader, A. T., & Al-Betar, M. A. "A survey on applications and variants of the cuckoo search algorithm". 2017. *Applied Soft Computing*.
- [5] Civicioglu, P., & Besdok, E. "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms". *Artificial intelligence review*, 1-32. 2013.
- [6] Yang, X.-S., & Deb, S. "Cuckoo search via Lévy flights. Paper presented at the Nature & Biologically Inspired Computing", 2009. NaBIC 2019.
- [7] Eberhart, R. C., & Kennedy, J. "A new optimizer using particle swarm theory". 1995.
- [8] V.K. Pathak, A.K. Singh, Optimization of morphological process parameters in contactless laser scanning system using modified particle swarm algorithm, *Measurement* (2017),
- [9] Yang, X.-S. "Firefly algorithms for multimodal optimization. Paper presented at the International Symposium on Stochastic Algorithms". 2019.
- [10] Chu, S.-C., Tsai, P.-W., & Pan, J.-S. "Cat swarm optimization". Paper presented at the Pacific Rim International Conference on Artificial Intelligence. 2016.
- [11] Yang, X.-S. "A new metaheuristic algorithm cooperative Nature strategy bat-inspired inspired for optimization" (NICSO 2010) (pp. 65-74): Springer. 2010b.
- [12] Dorigo, M., & Thomas, S. "Ant Colony Optimization". Cambridge, vol. 9, Dec. 20012: MIT Press. 2012.
- [13] Lang, Y.; Gao, Y. Dream Optimization Algorithm (DOA): A novel metaheuristic optimization algorithm inspired by human dreams and its applications to real-world engineering problems. *Comput. Methods Appl. Mech. Eng.* 436, 117718, 2025.
- [14] Dongshu Wang, Dapei Tan, Lei Liu, Particle swarm optimization algorithm: an overview, *Soft Comput.* 22 (2018) 387–408
- [15] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61.
- [16] Jiankai Xue, Bo Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, *Syst. Sci. Control Eng.* 8 (1) (2020) 22–34
- [17] Poomin Duankhan, Khamron Sunat, Sirapat Chiewchanwattana, Patchara Nasa-ngium, The Differentiated Creative search (DCS): Leveraging Differentiated knowledge-acquisition and Creative realism to address complex optimization problems, *Expert Syst. Appl.* (2024) 123734.
- [18] Ziyu Guan, Changjiang Ren, Jingtai Niu, Peixi Wang, Yizi Shang, Great Wall Construction Algorithm: A novel meta-heuristic algorithm for engineer problems, *Expert Syst. Appl.* 233 (2023) 120905

Corresponding author: Yakubu Abdulrazak

abdulrazakyakubu100@gmail.com

Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna.

© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved



- [19] Seyedali Mirjalili, Andrew Lewis, The whale optimization algorithm, *Adv. Eng. Softw.* 95 (2016) 51–67.
- [20] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019)," *IEEE Access*, vol. 9, pp. 26766-26791, 2021.
- [21] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge-based algorithm for solving optimization problems: a novel nature-inspired algorithm," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 7, pp. 1501-1529, 2020.
- [22] G. Atali, İ. Pehlivan, B. Gürevin, and H. Seker, Ibrahim., "Chaos in metaheuristic based artificial intelligence algorithms: a short review," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 29, no. 3, pp. 1354-1367, 2021.
- [23] A. T. Salawudeen, M. B. Mu'azu, A. Yusuf, and A. E. Adedokun, "A Novel Smell Agent Optimization (SAO): An extensive CEC study and engineering application," *Knowledge-Based Systems*, vol. 232, p. 107486, 2021.
- [24] A. T. Salawudeen et al., "Recent metaheuristics analysis of path planning optimization problems," in *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*, 2020: IEEE, pp. 1-7.
- [25] O. A. Meadows, M. B. Mu'Azu, and A. T. Salawudeen, "A Smell Agent Optimization Approach to Capacitated Vehicle Routing Problem for Solid Waste Collection," in *2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON)*, 2022: IEEE, pp. 1-5.
- [26] S. Vishnoi, S. Nikolovski, M. Raju, M. K. Kirar, A. S. Rana, and P. Kumar, "Frequency Stabilization in an Interconnected Micro-Grid Using Smell Agent Optimization Algorithm-Tuned Classical Controllers Considering Electric Vehicles and Wind Turbines," *Energies*, Vol. 16, no. 6, p. 2913, 2023.
- [27] A. T. Salawudeen, A. A. Olaniyan, G. A. Olarinoye, and T. H. Sikiru, "Formulation and Optimization of Overcurrent Relay Coordination in Distribution Networks Using Metaheuristic Algorithms," in *Information and Communication Technology and Applications: Third International Conference, ICTA 2020, Minna, Nigeria, November 24 27, 2020, Revised Selected Papers 3*, 2021: Springer, pp. 389-402.
- [28] S. Khan, S. Yang, and O. Ur Rehman, "A global particle swarm optimization algorithm applied to electromagnetic design problem," *International Journal of Applied Electromagnetics and Mechanics*, vol. 53, no. 3, pp. 451-467, 2017.
- [29] S. Khan, M. Kamran, O. U. Rehman, L. Liu, and S. Yang, "A modified PSO algorithm with dynamic parameters for solving complex engineering design problem," *International Journal of Computer Mathematics*, vol. 95, no. 11, pp. 2308-2329, 2018/11/02 2018, doi:10.1080/00207160.2017.1387252.
- [30] S. Mugemanyi *et al.*, "Marine predators algorithm : A comprehensive review," vol. 12, no. May, 2023.
- [31] M. Kumar, K. Rajwar, and K. Deep, "Analysis of Marine Predators Algorithm using BIAS toolbox and Generalized Signature Test," vol. 95, no. January, pp. 38–49, 2024.
- [32] T. Alabed and S. Servi, "A Lévy flight based chaotic black winged kite algorithm for solving optimization problems," pp. 1–28, 2025.
- [33] L. Wu, J. Wu, and T. Wang, *Enhancing grasshopper optimization algorithm (GOA) with levy flight for engineering applications*. 2023.
- [34] A. M. Z. WU LEI, WU JIAWEI, "Enhancing Grey Wolf Optimizer With Levy Flight for Engineering Applications," no. June, 2023.