Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A Novel Beagle Inspired Optimization Algorithm: Comprehensive Evaluation on Benchmarking Functions

Samindar Vibhute[1*], Chetan Arage[2†]
[1]Computer Science and Engineering, Sanjay Ghodawat University, Atigre, Kolhapur, 416118, Maharashtra, India.ORCID: 0000-0001-5797-0874*
[2]Computer Science and Engineering, Sanjay Ghodawat University, Atigre, Kolhapur, 416118, Maharashtra, India.
**Corresponding author(s). E-mail(s): samindarvibhute1@gmail.com;*
Contributing authors: chetan.arage@sanjayghodawatuniversity.ac.in;
[†]These authors contributed equally to this work.

ARTICLE INFO

Received: 10 Oct 2024

Revised: 25 Nov 2024

Accepted: 26 Dec 2024

ABSTRACT

This paper presents an evaluation of the novel Beagle-Inspired Optimization Algorithm (BIOA), inspired by the scent detection and rabbit hunting strategies of beagle dogs, such as scent detection, tracking, trail following, pattern recognition, continuous adaptation, persistent and exhaustive search, and escape and retrieval. BIOA is compared with well-established algorithms, including Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), and Cuckoo Search (CS), across a set of benchmark functions, including Sphere, Rosenbrock, Rastrigin, Griewank, Ackley, Levy, and Schwefel functions. The results demonstrate BIOA's superior performance, achieving the lowest mean fitness values and best solutions across most test cases. Its balanced exploration and exploitation phases enable effective optimization. While BIOA excels in many instances, it requires further improvements in computational efficiency, particularly for high-dimensional problems. Future research should focus on enhancing BIOA's performance through advanced models, hybrid optimization techniques, and real-world problem applications, thus broadening its practical impact in solving complex optimization tasks.

Keywords: Beagle-Inspired Optimization Algorithm, Benchmark Functions, Nature Inspired Algorithms, Exploration and Exploitation, Performance Evaluation

Abbreviations

BIOA: Beagle-Inspired Optimization Algorithm

PSO: Particle Swarm Optimization

ABC: Artificial Bee Colony

ACO: Ant Colony Optimization

CS: Cuckoo Search

1 Introduction

Optimization algorithms are pivotal in addressing a broad spectrum of complex problems in diverse fields, such as machine learning, engineering, finance, healthcare, and logistics [1]. These algorithms iteratively search for the best possible solution within a defined search space, aiming to find the global optimum. While numerous optimization techniques have emerged, nature-inspired algorithms have gained widespread attention due to their ability to mimic the adaptability, efficiency, and robustness of natural systems. Among these, evolutionary and swarm-based algorithms, such as PSO [2], ACO [3], ABC [4], and CS [5], have proven to be highly effective in solving complex optimization problems. These algorithms are designed based on the collective intelligence observed in nature, particularly the behaviors of animals and insects, and they offer solutions to problems where traditional methods may struggle [6].

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

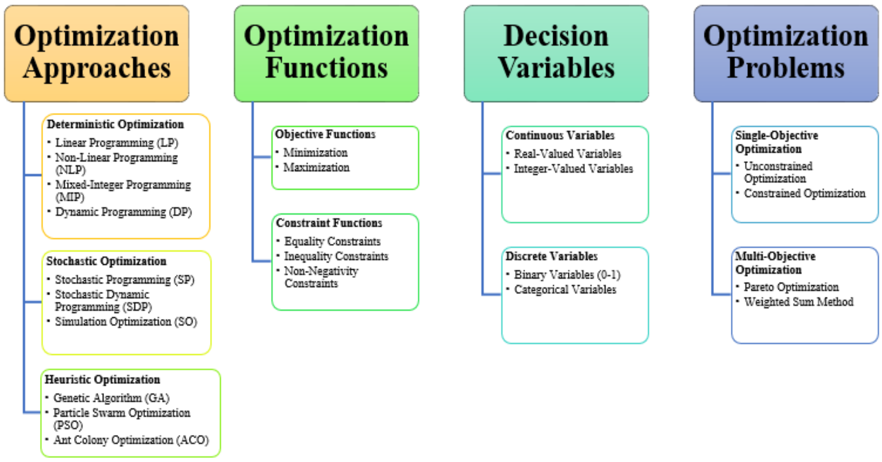

Figure 1Classification of Optimization

Deterministic optimization methods presume that the parameters of the optimization challenge are completely understood [7]. These methods consist of linear programming, non-linear programming, mixed-integer programming, and dynamic programming as shown in *Figure 1*. For instance, linear programming is employed to maximize linear objective functions subject to linear constraints. It is commonly utilized in operations research and management science. Non-linear programming, in contrast, is utilized to optimize non-linear objective functions alongside either linear or non-linear constraints. It is applied across different domains, such as engineering, economics, and finance.

In contrast, stochastic optimization methods presume that the optimization problem's parameters are uncertain or random [8]. These methods encompass stochastic programming, stochastic dynamic programming, and simulation optimization. Stochastic programming, for instance, is utilized to enhance performance amidst uncertainty by employing probability distributions. It finds application in multiple areas, such as finance, economics, and engineering. In contrast, simulation optimization employs simulation models to enhance systems in the face of uncertainty. It is utilized in multiple areas, such as logistics, supply chain management, and finance.

Heuristic optimization techniques [9] utilize approximation methods to discover near-optimal solutions. These methods consist of genetic algorithms, PSO, ACO, and simulated annealing. Genetic algorithms, for instance, draw inspiration from natural selection and genetics to find the best solutions. PSO imitates the behavior of flocks of birds or schools of fish to find optimal solutions [2]. ACO mimics the searching behavior of ants to discover the best routes [3]. Simulated annealing employs a temperature sequence to prevent local optima and discover global optima.

Optimization functions are essential in optimization challenges. These functions can be divided into objective functions and constraint functions [10]. Objective functions can be functions of either minimization or maximization. Minimization functions seek to identify the lowest value of the objective function, whereas maximization functions strive to discover the highest value [11]. In contrast, constraint functions may be categorized as equality constraints, inequality constraints, and non-negativity constraints. Equality constraints need to be met precisely, whereas inequality constraints should be fulfilled within a specific range. Non-negativity restrictions guarantee that variables remain non-negative.

Decision variables are crucial in optimization problems, particularly in the context of Many Objective Optimization Problems (MaOPs), where traditional algorithms often struggle with scalability. The Decision Variable Learning (DVL) algorithm, which employs machine learning to predict solutions close to the Pareto-optimal front, has demonstrated superior performance compared to established methods like NSGA-III [12,13]. Additionally, the Subset Selection Algorithm (SSA) effectively reduces the complexity of decision variable combinations, enhancing resource efficiency in complex chemical processes [14]. Furthermore, understanding interactions between decision variables is essential, as these interactions can significantly influence outcomes in engineering and classification tasks. Techniques such as distributed decision variable analysis and Exploratory Landscape Analysis (ELA) are being developed to manage large-scale decision variables and their interactions, thereby improving optimization efficiency [15].

Optimization issues can be divided into single-objective optimization issues and multi-objective optimization issues. Single-objective optimization problems focus on optimizing one objective function, whereas multi-objective optimization problems concentrate on optimizing several objective functions at the same time [16]. Single-objective optimization issues can additionally be divided into unconstrained optimization issues and constrained optimization issues. Unconstrained optimization problems lack constraints, whereas constrained optimization problems include restrictions that must be met. Various methods can be employed to solve multi-objective optimization problems, such as the weighted sum method and Pareto optimization. The weighted sum approach allocates weights to every objective function and seeks to optimize the total weighted sum [17]. In contrast, Pareto optimization identifies the Pareto front, where improving one objective function would lead to the deterioration of another.

Despite the success of current bio-inspired algorithms, there remains an ongoing need for novel approaches that can effectively balance the exploration and exploitation phases of optimization, while also maintaining adaptability to diverse problem landscapes [18]. In this context, the BIOA is proposed, which takes inspiration from the scent detection and hunting strategies of Beagles, coupled with the adaptive escape and retrieval tactics observed in rabbits. Beagles, known for their remarkable scent-detection abilities, track scents over long distances by employing a methodical, persistent, and exhaustive search strategy [19]. Similarly, rabbits utilize highly adaptive strategies to escape predators, demonstrating an ability to adapt and persist in challenging environments [20]. By integrating these characteristics—persistent search, tracking, pattern recognition, adaptation, and escape—BIOA is designed to enhance the efficiency of optimization algorithms, especially for complex, multimodal, and high-dimensional problems.

BIOA aims to address some of the shortcomings of existing algorithms by providing an optimization process that adapts more flexibly to the search landscape. The key inspiration for BIOA lies in the remarkable tracking and scent-following behavior of Beagles and the rabbit's behavioral strategies, including continuous adaptation and escape. BIOA is intended to enhance the balance between exploration (searching for global optima) and exploitation (refining known good solutions) within the optimization search process. It emphasizes persistent exploration, adapting its search patterns based on the best solutions found while incorporating escape mechanisms to avoid local optima, akin to the way animals adapt to dynamic environments and threats.

This paper introduces the BIOA as a novel optimization framework and assesses its performance on benchmark functions, comparing it with well-established optimization techniques such as PSO, ABC, ACO, and CS. We aim to demonstrate how BIOA outperforms other algorithms in terms of solution quality, computational efficiency, and robustness across diverse problem landscapes. The results indicate that BIOA provides significant improvements, especially in solving high-dimensional and multimodal optimization problems. Additionally, we explore the potential real-world applications of BIOA, emphasizing its suitability for complex optimization tasks, including those found in fields such as machine learning, robotics, and logistics. This research contributes to the growing body of knowledge in bio-inspired optimization algorithms and presents new possibilities for future developments in nature-based computational techniques.

The primary motivation for BIOA is to offer a more versatile optimization technique capable of effectively navigating complex solution spaces while maintaining flexibility and adaptability. Future work will focus on improving BIOA's robustness, exploring hybrid approaches with other bio-inspired algorithms, and testing its performance on real-world applications. The study's findings point to the exciting potential of BIOA to revolutionize the landscape of optimization algorithms, pushing the boundaries of what is possible in computational optimization.

2 Literature Review

Optimization algorithms have evolved significantly over the past several decades, addressing increasingly complex problems across diverse fields such as engineering, machine learning, robotics, and logistics. In traditional optimization methods, algorithms such as gradient descent and Newton's method were used to find local optima [21]. However, these methods often fail when dealing with multimodal, high-dimensional, or non-differentiable objective functions. To address these challenges, bio-inspired optimization techniques have gained widespread attention.

These algorithms mimic natural processes and behaviors of living organisms, combining efficiency and robustness to solve complex optimization problems [22].

One of the most well-known bio-inspired algorithms is PSO [23], first proposed by [36]. PSO is inspired by the social behavior of birds flocking or fish schooling, where each particle (solution) adjusts its position based on its own experience and that of its neighbors. PSO has proven successful in solving a wide range of optimization problems, particularly in continuous optimization tasks. However, its main limitation lies in its tendency to converge prematurely to local optima, particularly in multimodal and high-dimensional spaces [24].

ACO is another widely used bio-inspired optimization algorithm. Inspired by the foraging behavior of ants, ACO uses a decentralized approach where ants communicate indirectly through pheromones. This method allows for the exploration of the solution space and has been particularly successful in combinatorial optimization problems like the traveling salesman problem [25]. Although ACO is effective in many applications, it can be computationally expensive and requires fine-tuning of parameters such as pheromone evaporation rates [26].

ABC is another algorithm that draws inspiration from nature, specifically the foraging behavior of honey bees. In ABC, employed bees search for nectar sources and share information with onlooker bees, allowing the algorithm to balance exploration and exploitation [27]. The algorithm has demonstrated high efficiency in solving continuous optimization problems. However, ABC faces similar challenges as other bio-inspired algorithms, such as slow convergence in high-dimensional and multimodal problems [28].

CS, studied by Reda [29], is based on the brood parasitism behavior of cuckoo birds, where they lay their eggs in the nests of other birds, forcing the host to adopt their offspring. The CS algorithm employs a simple, yet efficient strategy of random walks and the Lévy flight mechanism to explore the solution space. While CS has shown promising results, particularly in high-dimensional optimization problems, it still faces challenges in terms of its computational complexity and convergence speed [30].

Although these algorithms have achieved success in various optimization tasks, there is still a need for novel approaches that balance the exploration and exploitation phases effectively. The BIOA aims to address these challenges by drawing inspiration from the scent detection abilities of Beagles and the hunting strategies of rabbits. Beagles are renowned for their remarkable ability to track scents over long distances, using a methodical, persistent, and exhaustive search strategy [31]. Similarly, rabbits demonstrate adaptability in their escape tactics, evading predators through continuous adaptation and persistent efforts [32]. BIOA integrates these traits—scent detection, pattern recognition, continuous adaptation, and exhaustive search strategies—to offer a novel approach to optimization. The incorporation of such adaptive behaviors from animals could enhance the performance of optimization algorithms, particularly in dealing with complex, multimodal, and high-dimensional landscapes [31].

The growing body of research on bio-inspired algorithms suggests that these methods can be effectively combined to achieve more robust optimization strategies. Hybridization of multiple algorithms, for example, combining PSO and ACO, has been explored to improve convergence speed and solution accuracy [32]. Additionally, studies have shown that incorporating mechanisms such as dynamic adaptation and self-learning could further enhance the robustness of optimization algorithms [33, 34]. These hybrid approaches, along with the design of more sophisticated algorithms like BIOA, offer new opportunities for solving complex optimization problems in real-world scenarios.

Table 1 Overview of existing algorithms

Algorithm	Inspiration	Advantages	Challenges	Applications
Particle Swarm Optimization (PSO)	Social behavior of birds flocking, fish schooling	Simple, easy to implement; good balance of exploration and exploitation	Tendency to converge prematurely; difficulty in high-dimensional problems	Continuous optimization, neural network training, feature selection

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Ant Colony Optimization (ACO)	Foraging behavior of ants	Effective in combinatorial problems; decentralized, robust against failures	Computationally expensive; sensitive to parameter tuning	Traveling Salesman Problem, vehicle routing, network design
Artificial Bee Colony (ABC)	Foraging behavior of honey bees	Balances exploration and exploitation; easy to implement	Slow convergence in high-dimensional spaces; sensitive to population size	Continuous optimization, function optimization, data clustering
Cuckoo Search (CS)	Brood parasitism behavior of cuckoo birds	Efficient for high-dimensional optimization; uses Lévy flights for exploration	Computationally expensive; requires fine-tuning	Function optimization, engineering design, feature selection
Beagle-Inspired Optimization Algorithm (BIOA)	Scent detection and hunting strategies of Beagles and rabbits	Persistent and exhaustive search; continuous adaptation and pattern recognition	Requires further research to optimize performance in complex, high-dimensional problems	Multi-modal optimization, real-world problem solving

Optimization algorithms are mathematical tools used to find the best solution to a problem by navigating a solution space. These algorithms *Table 1* are widely applied in various fields, including machine learning, engineering design, economics, and operations research [35]. The effectiveness and efficiency of these algorithms are highly influenced by the parameters that control their behavior, such as population size, exploration/exploitation balance, and the termination criteria. Understanding and fine-tuning these parameters are essential for improving the performance of optimization algorithms, ensuring convergence to optimal or near-optimal solutions while minimizing computational costs.

Parameters such as population size, learning factors, step size, and iteration limits play a crucial role in determining the behavior of algorithms. For example, the population size affects the diversity of solutions, with larger populations often leading to a more exhaustive search of the solution space, but at the expense of higher computational costs [36]. The balance between exploration and exploitation is another critical factor, as algorithms must explore enough of the search space to avoid local optima, while also exploiting known good solutions to find the global optimum [37]. In the case of swarm intelligence-based algorithms like PSO and ACO, parameters like the cognitive and social factors PSO or pheromone evaporation rate ACO can significantly impact their ability to converge efficiently [38].

In addition, the step size or velocity parameters, such as those in PSO or CS, influence how aggressively the algorithm explores the search space. A larger step size leads to faster exploration, while a smaller one can enable more precise refinement [39]. The termination criteria, such as iteration count or fitness threshold, determine when the algorithm should stop, which can influence both the quality of the solution and the time taken to reach it.

The necessity to study the parameters of optimization algorithms arises from their direct influence on algorithmic performance. The choice of parameters can dramatically alter the convergence rate, the quality of the final solution, and the computational efficiency of the algorithm. By carefully tuning these parameters, researchers can enhance algorithm performance, allowing for more efficient problem-solving across a variety of domains. A deep understanding of these parameters helps in determining the optimal settings for specific problem types, ultimately leading to the development of more efficient, adaptive, and robust optimization techniques [40].

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

In addition, as real-world optimization problems become increasingly complex and large-scale, the ability to customize parameters to better suit these problems becomes more critical. For example, algorithms like the ABC and CS have several parameters that govern the exploration and exploitation process. Understanding how these parameters influence the algorithm's behavior helps in designing more effective hybrid optimization techniques, which can combine strengths from different algorithms to solve complex problems with higher accuracy and efficiency [41].

Table 2Key Parameters of PSO, ACO, ABC, and CS Algorithms

Algorithm	**Key Parameters**	**Description**
PSO	Population Size (N)	The number of particles in the swarm. A larger population increases the exploration of the search space.
	Cognitive Parameter (c_1)	Determines the pull of each particle towards its own best-known position.
	Social Parameter (c_2)	Determines the pull of each particle towards the global best-known position.
	Inertia Weight (w)	Controls the trade-off between exploration and exploitation (larger w favors exploration).
	Velocity Limits (V_max)	Defines the maximum velocity a particle can move. Affects convergence rate.
	Number of Iterations (T)	The total number of iterations or steps for the algorithm to run.
ACO	Number of Ants (N)	The total number of ants used in the search process. Affects exploration and robustness of the solution.
	Pheromone Evaporation Rate (ρ)	The rate at which the pheromone trails decay over time. Affects the exploitation of solutions.
	Pheromone Influence (α)	The importance of the pheromone trail in the decision-making process.
	Visibility (β)	The importance of the problem's heuristic information (visibility).
	Number of Iterations (T)	The total number of iterations or cycles for the algorithm to run.
ABC	Number of Bees (N)	The number of employed bees and scout bees in the colony. Affects the algorithm's exploration and exploitation.
	Discovery Rate (Φ)	Probability of a scout bee searching for a new solution. Influences the exploration phase.
	Onlooker Bees (O)	The number of bees that observe and select the solutions based on probability.

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

	Limit (L)	Maximum number of cycles without improvement before a bee becomes a scout.
	Number of Iterations (T)	The total number of iterations or cycles for the algorithm to run.
CS	Population Size (N)	The number of cuckoos in the population. Affects diversity and exploration.
	Step Size (α)	The step size (also called the step length) used during the Lévy flights for exploration.
	Pa (Abandonment Probability)	The probability that a cuckoo's egg is abandoned by the host bird. Influences exploration and exploitation.
	Lévy Flight Distribution	Defines the movement behavior of the cuckoos during their random walk.
	Number of Iterations (T)	The total number of iterations or steps the algorithm runs

The study of optimization algorithm parameters *Table 2* is essential for understanding and improving the performance of these algorithms. By fine-tuning parameters such as population size, exploration/exploitation balance, and velocity/step size, researchers can develop algorithms that converge faster, find better solutions, and operate more efficiently in real-world applications. This research area continues to evolve, with new insights into parameter influence enabling the development of more sophisticated and powerful optimization algorithms.

3 Proposed Algorithm

3.1 Inspiration

Beagles are highly specialized scent hounds known for their remarkable olfactory capabilities and persistent hunting behaviors, which have inspired the design of the BIOA. The key biological traits of Beagles that serve as inspiration for the algorithm include their exceptional scent detection, trail-following techniques, pattern recognition, adaptability, and persistence. Beagles possess an extraordinary sense of smell, allowing them to detect faint scent trails and follow them with remarkable accuracy. They continuously analyze the air and adjust their position to hone in on the source of the scent, which parallels how an optimization algorithm searches for promising solutions within a complex search space.

Once a scent is detected, Beagles exhibit a dynamic tracking behavior, often moving in zigzag or circular patterns rather than following a direct path. This non-linear approach helps them refine their trajectory, ensuring they stay on the freshest scent trail, much like an algorithm balances exploration and exploitation to find an optimal solution. Furthermore, Beagles have an innate ability to recognize patterns within scent trails, distinguishing between fresh and older scents. This trait mirrors the algorithm's heuristic evaluation, where it differentiates between promising and less viable solutions, focusing resources on the most promising areas.

Adaptability is another crucial trait of Beagles that influences the algorithm design. If an obstacle blocks the scent trail, a Beagle instinctively recalculates its path, exploring alternative routes to regain the trail. Similarly, an optimization algorithm must be capable of dynamically adjusting its search trajectory when faced with barriers, ensuring robustness in solving complex problems. Additionally, Beagles exhibit relentless persistence in their pursuit, covering large areas and persisting in challenging terrains until they locate their target. This persistence is an essential aspect of BIOA, ensuring thorough exploration of the search space without premature convergence.

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

The inspiration drawn from Beagles extends to their efficient decision-making mechanisms. When a Beagle detects a change in the scent's direction or strength, it swiftly repositions itself to stay on track, which reflects the algorithm's need to perform dynamic updates and recalibrate search paths to avoid getting stuck in suboptimal solutions. Overall, the unique biological characteristics of Beagles scent detection, adaptive trail-following, pattern recognition, and persistence serve as a robust foundation for the BIOA, making it a powerful tool for solving complex optimization problems in dynamic environments.

3.2 Mathematical Model

3.2.1 **Scent Detection (Initialization)**

Objective: Randomly initialize agents' positions and evaluate their fitness.

- **Initialize:**

$$X_i^t \sim U(0,1)^D, i = 1,2,3, \dots, N$$

- **Evaluate fitness:**

$$f_i^t = f(X_i^t), i = 1,2,3 \dots, N$$

- **Determine the best solution:**

$$X_{Best}^t = arg \min_i f_i^t, f_{Best}^t = \min_i f_i^t$$

Figure 2 Scent Detection by Beagle

3.2.2 **Tracking/Trail Following (Exploration Phase)**

Objective: Perform a random walk to explore new regions in the search space.
For each agent i.

- **Update Position:**

$$X_i^{t+1} = X_i^t + \Delta X, \Delta X \sim U(-1,1)^D$$

- **Evaluate fitness:**

$$f_i^{t+1} = f(X_i^{t+1})$$

- **Update agent position if fitness improves:**

$$If\ f_i^{t+1} < f_i^t, X_i^t \leftarrow X_i^{t+1}, f_i^t \leftarrow f_i^{t+1}$$

- **Update global best if necessary:**

$$If\ f_i^{t+1} < f_{Best}^t, \quad X_{Best}^t \leftarrow X_i^t, \quad f_{Best}^t \leftarrow f_i^t$$

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

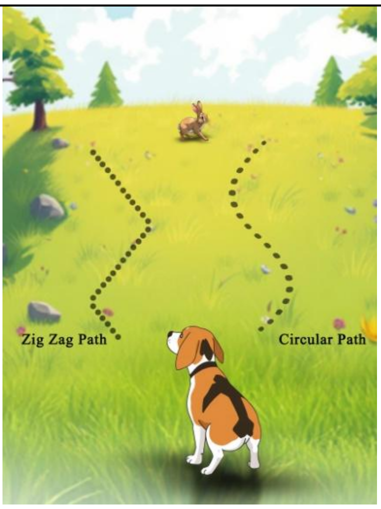

Figure 3Tracking & Trail Following

3.2.3 Pattern Recognition (Exploitation Phase)

Objective: Exploit the best solution by moving agents closer toX_{Best}
For each agent i

- **Update position:**

$$X_i^{t+1} = X_i^t + U(-1,1)^D.(X_{Best}^t - X_i^t)$$

- **Evaluate fitness:**

$$f_i^{t+1} = f(X_i^{t+1})$$

- **Update agent position if fitness improves:**

$$If\ f_i^{t+1} < f_{i,}^t \qquad X_i^t \leftarrow X_i^{t+1}, \qquad f_i^t \leftarrow f_i^{t+1}$$

- **Update global best if necessary:**

$$If\ f_i^t < f_{Best,}^t\ X_{Best}^t \leftarrow X_i^t, \qquad f_{Best}^t \leftarrow f_i^t$$

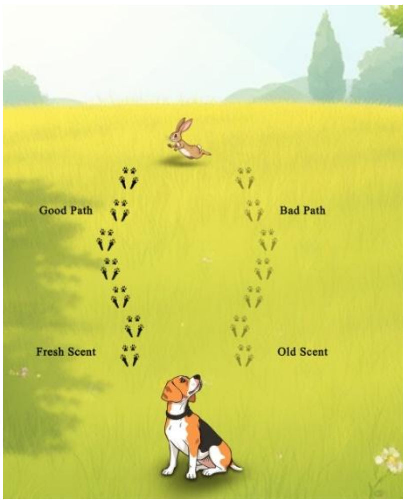

Figure 4Pattern Recognition

3.2.4 **Continuous Adaptation (Adaptation Phase)**

Objective: Reinitialize some agents randomly to avoid local optima.
For each agent i:

- **Random re-initialization with probability P_{Adapt}:**

$$If\ r < P_{Adapt}, \qquad X_i^t \sim U(0,1)^D$$

- **Re-evaluate fitness:**

$$f_i^t = f(X_i^t)$$

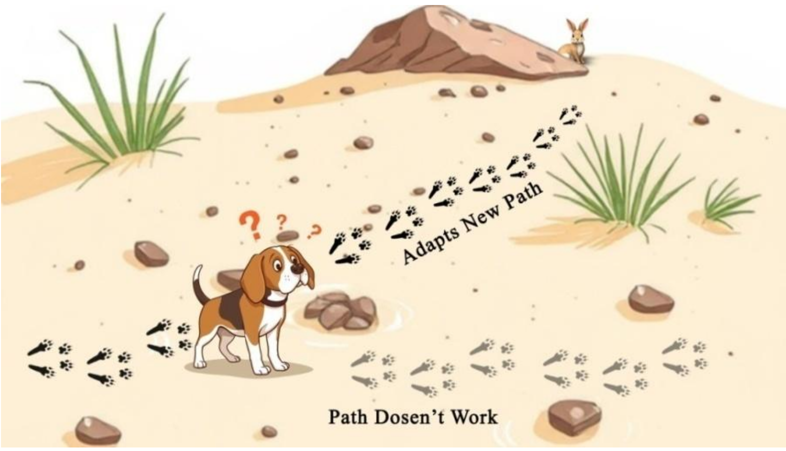

Figure 5Continuous Adaption by Beagle

3.2.5 **Persistent & Exhaustive Search (Termination Check)**

Objective: Check if termination criteria are met.

- **Terminate if:**

$$t = T_{Max} \text{or other convergence criteria}$$

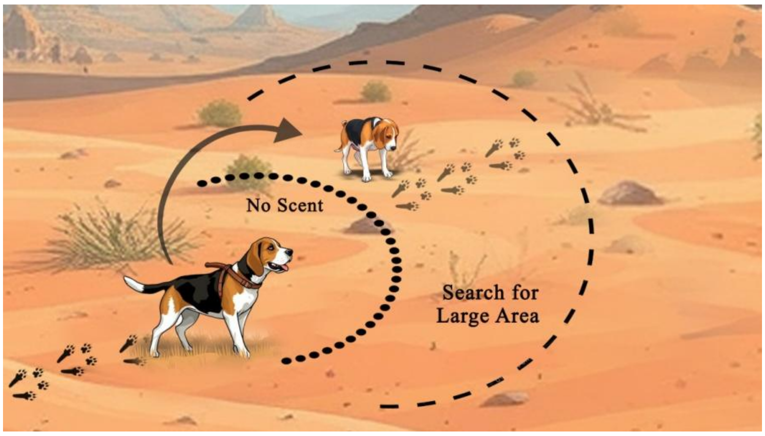

Figure 6Persistent & Exhaustive Search

3.2.6 **Escape & Retrieval (Repeat Process)**

Objective: Continue the optimization until termination criteria are met.

- **Repeat phases (2) through (5) until$t = T_{Max}$**

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- **Final Best Solution:**

$$X_{Best}, \quad f_{Best}$$

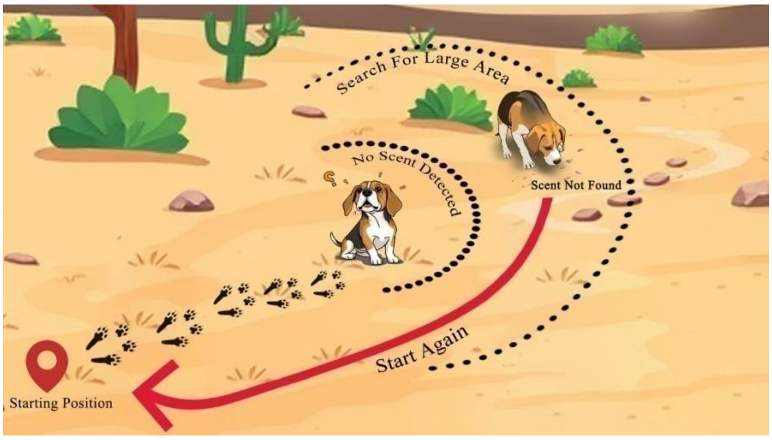

Figure 7Escape & Retrieval by Beagle

3.3 Flowchart

The BIOA follows a structured sequence of operations *Figure 8 BIOA Flowchart* inspired by the scent-tracking behavior of Beagles. The process begins with the initialization phase, where the algorithm is set up by defining key parameters such as the population size, the dimensions of the search space, and the maximum number of iterations. This step is crucial as it establishes the foundation for the algorithm's search process by positioning the agents, representing Beagles, randomly within the problem domain to ensure diverse exploration.

Once the initialization is complete, the algorithm proceeds to the fitness evaluation phase, where each agent assesses the quality of its current position using a predefined fitness function. This function quantifies the proximity of a given solution to the optimal solution, allowing agents to determine their effectiveness in the search space. If the best possible solution is identified at this stage, the algorithm terminates, outputting the optimal solution. However, if an optimal solution is not yet found, the algorithm moves to the core search phases.

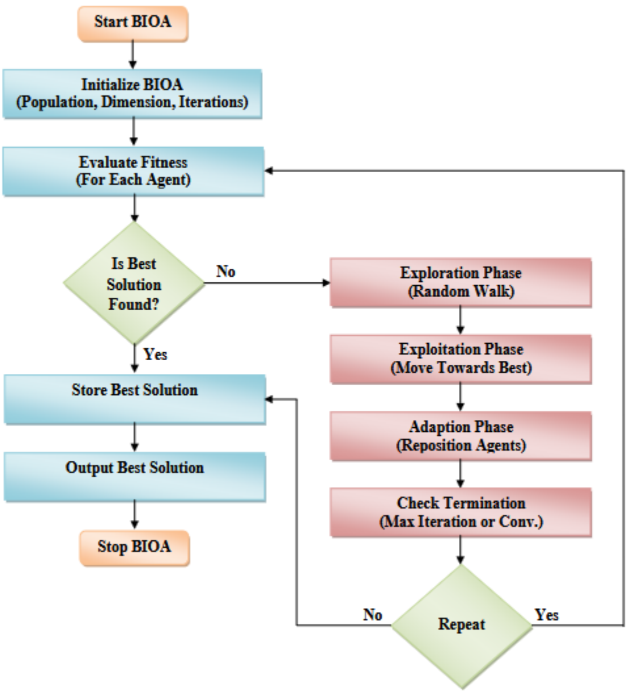

Figure 8 BIOA Flowchart

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

The search process consists of three primary phases: exploration, exploitation, and adaptation. In the exploration phase, agents perform a random walk, mimicking the behavior of Beagles that scan large areas for scent trails. This phase promotes diversity in the search process by enabling agents to investigate unexplored regions and avoid becoming trapped in local optima. Following exploration, the exploitation phase directs agents to move towards the best solutions found so far, simulating how a Beagle intensifies its pursuit once a promising scent trail is detected. This phase ensures that the algorithm focuses on refining high-potential solutions while continuing to balance exploration.

The adaptation phase is a crucial aspect of BIOA, allowing agents to reposition themselves when they encounter obstacles or weak solutions. Inspired by the way Beagles recalibrate their path when a scent weakens or is obstructed, this phase helps the algorithm dynamically adjust its search trajectory, improving robustness and flexibility in dynamic environments.

Throughout the search process, the algorithm continuously checks termination conditions. The process halts when either the maximum number of iterations is reached or convergence criteria are met, indicating that agents have collectively settled on an optimal solution. If neither condition is satisfied, the search process repeats by re-evaluating the fitness of agents and iterating through the exploration, exploitation, and adaptation phases.

Finally, upon termination, the algorithm outputs the best solution identified during the search, presenting the optimal solution along with its corresponding fitness value. This structured flow ensures that BIOA efficiently balances exploration and exploitation, adapts dynamically to changing search conditions, and persistently seeks optimal solutions across a diverse range of optimization problems.

3.4 Pseudo code for Beagle Algorithm

```
1. Initialize agents randomly.
2. Evaluate fitness for all agents.
3. Find the best solution and best fitness.
4. Repeat until max_iterations:
   a. Exploration Phase:
      - Move agents randomly.
      - Evaluate new fitness and update agents if needed.
   b. Exploitation Phase:
      - Move agents toward the best-known solution.
      - Evaluate new fitness and update agents if needed.
   c. Adaptation Phase:
      - With small probability, reinitialize agents.
5. Check termination condition.
6. Return best solution and best fitness.
```

The algorithm begins with the Initialization Phase, where a group of agents (representing beagles) is randomly distributed in the search space. Each agent's position corresponds to a potential solution to the optimization problem, and its fitness value is calculated using the given objective function. The algorithm identifies the best solution among these initial agents by finding the position with the minimum fitness value, which will act as the benchmark for the following iterations.

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Next, the algorithm enters a loop that continues until a predefined number of iterations is reached or a termination condition is satisfied. This loop consists of three primary phases: Exploration, Exploitation, and Adaptation, which emulate the behaviors of beagles in a scent-tracking scenario.

In the Exploration Phase, the agents perform a random walk in the search space, simulating the random sniffing of beagles as they try to locate the scent trail. Each agent's new position is calculated by adding a random vector to its current position. The fitness of the new position is evaluated, and the agent updates its position if the new position offers an improvement. If any agent finds a better solution than the current global best, the algorithm updates the global best solution and its fitness.

The Exploitation Phase follows, during which agents focus their search around the best solution found so far. Each agent moves closer to the global best position by adjusting its movement vector to be weighted toward the best solution. This phase mimics the beagle's behavior of converging on a strong scent trail. As in the exploration phase, the fitness of each new position is evaluated, and the agents update their positions if the new position improves their fitness. If any agent discovers a better solution than the global best, the global best solution is updated.

To maintain diversity and avoid premature convergence, the algorithm includes an Adaptation Phase. During this phase, each agent has a small probability of being reinitialized to a new random position in the search space. This process ensures that the algorithm explores unexplored regions of the search space, thereby increasing the likelihood of escaping local optima. The fitness of these reinitialized agents is also recalculated.

At the end of each iteration, the algorithm performs a Termination Check to determine whether the maximum number of iterations has been reached or if another convergence criterion has been satisfied. If neither condition is met, the algorithm loops back to the exploration phase and continues searching. This persistent and exhaustive search mimics the relentless nature of beagles as they hunt until they locate their target or exhaust all options.

Finally, once the termination condition is met, the algorithm returns the best solution and its corresponding fitness value. These represent the optimal or near-optimal solution to the optimization problem. This structured combination of exploration, exploitation, and adaptation phases ensures a balance between global and local search, making BIOA both robust and versatile for solving various optimization problems.

4 Experimental Setup

4.1 Benchmark Function

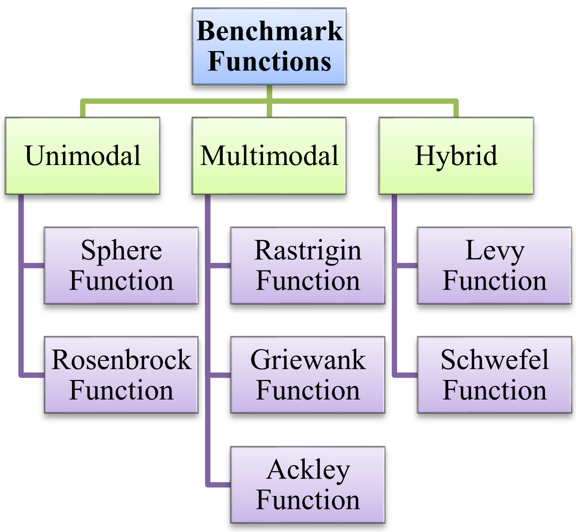

Figure 9Benchmark Functions Tree

Benchmark functions are critical tools in optimization and computational intelligence research, serving as standardized test problems to evaluate the performance of optimization algorithms. These *Figure 9* functions are broadly categorized based on their landscape and characteristics, such as unimodal, multimodal, and hybrid.

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

Unimodal functions feature a single global optimum, making them ideal for testing the exploitation capabilities of optimization algorithms. The Sphere Function, one of the simplest unimodal functions, represents a quadratic bowl-shaped surface where the global minimum lies at the origin. Its smooth and symmetric landscape allows researchers to test an algorithm's ability to converge efficiently. Another widely used unimodal function is the Rosenbrock Function, often referred to as the "banana function" due to its curved valley shape. It is more challenging than the Sphere Function, as the global minimum lies inside a narrow, elongated parabola, testing an algorithm's precision in finding the optimum.

Multimodal functions are characterized by multiple local optima, making them suitable for evaluating an algorithm's exploration capabilities. The Rastrigin Function is a classic example, combining a parabolic shape with a sinusoidal modulation that creates numerous local minima. The Griewank Function, on the other hand, is more complex due to its product and summation terms, which introduce periodic variations, further challenging an algorithm's robustness. Similarly, the Ackley Function presents a flat outer region with a sharp central peak, requiring algorithms to balance exploration and exploitation to locate the global minimum.

Table 3Benchmark Function, Formula, Range

Sr. No.	Function Name	Formula	Range
1	**Sphere Function**	$f(x) = \sum_{i=1}^{n} x_i^2$	$x_i \epsilon[-5.12,5.12]$
2	**Rosenbrock Function**	$f(x) = \sum_{i=1}^{n} [100(x_{i\|1} - x_i^2)^2 + (x_i - 1)^2]$	$x_i \epsilon[-2.048,2.048]$
3	**Rastrigin Function**	$f(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)]$	$x_i \epsilon[-5.12,5.12]$
4	**Griewank Function**	$f(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \pi_{i=1}^{n}.\cos\left(\frac{x_i}{\sqrt{i}}\right)$	$x_i \epsilon[-600,600]$
5	**Ackley Function**	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$x_i \epsilon[-32.768,32.7$
6	**Levy Function**	$f(x) = sin^2(\pi w_1) + \sum_{i=1}^{n}(w_1 - 1)^2[1 + 10sin^2(\pi w_i + 1) + (w_n - 1)^2[1 + sin^2(2\pi w_n)]]$ $where\ w_i = 1 + \frac{x_i.1}{100}$	$x_i \epsilon[-10,10]$
7	**Schwefel Function**	$f(x) = 418.9829d - \sum_{i=1}^{n} x_i \sin(\sqrt{\|x_i\|})$	$x_i \epsilon[-500,500]$

Hybrid functions combine characteristics of different function types, often incorporating multiple components or transformations to create challenging optimization problems. The Levy Function features a combination of multimodal and deceptive properties, with its global minimum surrounded by numerous local optima, testing an algo-

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ Research Article

rithm's ability to escape suboptimal solutions. Similarly, the Schwefel Function has a highly irregular landscape with steep ridges and deep valleys, making it particularly difficult for optimization algorithms to navigate effectively.

Overall, these benchmark functions *Table 3* provide diverse landscapes and challenges, enabling comprehensive evaluation of optimization algorithms in terms of convergence speed, accuracy, and robustness.

4.2 Experimental Parameters

In optimization algorithms, several parameters define the behavior, performance, and effectiveness of the search process. These parameters *Table 4* play a critical role in balancing exploration (global search) and exploitation (local search) and ensuring the algorithm converges to an optimal solution.

The Number of Agents (N) refers to the total number of agents (e.g., beagles in nature-inspired algorithms) that participate in searching the solution space. Typically, the number of agents ranges from 20 to 100, with a common default value of 50. Increasing N enhances exploration but adds computational overhead. The Number of Dimensions (D) represents the problem's variables or the dimensionality of the search space, which depends on the specific problem being addressed. Typical values for D are 2, 10, or 30, but this may vary depending on the complexity of the optimization problem.

Table 4 BIOA Parameters

Parameter	Description	Typical Values/Range
Number of Agents($\boldsymbol{N}$)	The total number of agents (beagles) searching for the optimal solution.	$20 \leq N \leq 100\ (e.g.50)$
Number of Dimensions($\boldsymbol{D}$)	The dimensionality of the search space, representing the problem's variables.	$Dependsontheproblem,$ $e.g.2, 10\ or\ 30$
Objective Function($\boldsymbol{f}(\boldsymbol{X})$)	The fitness function to evaluate the quality of solutions.	Custom problem specific function.
Max Iterations$\boldsymbol{T_{Max}}$	Maximum number of iterations before stopping the optimization process.	$100 \leq T_{Max} \leq 1000$ $e.g.100$
Exploration Range ($\boldsymbol{U(-1,1)}$)	The random step size for exploration in the search space.	$UniformDistribution$, $[-1,1]$
Exploitation Factor($\boldsymbol{C}$)	Weighting factor for exploitation phase to move toward the best solution.	$0.1 \leq C \leq 1.0$ $(Default\ 1)$
Adaptation Probability$\boldsymbol{P_{Adapts}}$	Probability of an agent randomly repositioning itself in the search space.	$0.05 \leq P_{Adapt} \leq 0.2$ $e.g.0.1$
Initial Positions$\boldsymbol{X_i^0}$	Initial random positions of agents in the search space.	$Randomvalues.$ $X_i \sim U(0,1)^D$
Best Solution$\boldsymbol{X_{Best}}$	Position of the agent with the best fitness in the current	Updated dynamically during optimization

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

	population.	
Fitness of Best Solution(f_{Best})	Fitness value of the best solution found so far.	Updated dynamically during optimization
Random Step(ΔX)	The random offset added to agents' positions in exploration and exploitation phases.	Uniform Random values $[-1,1]^D$
Convergence Threshold(ϵ)	Fitness threshold for stopping early if the solution is close enough to optimal.	Small values $e.g. 10^6$

The **Objective Function** $(\boldsymbol{f}(\boldsymbol{X}))$serves as the fitness function to evaluate the quality of each solution. This function is customized for the problem at hand, determining whether a solution is closer to optimal. The **Maximum Number of Iterations Max Iterations**$\boldsymbol{T_{Max}}$is the predefined limit for the optimization process, with typical values ranging between 100 and 1000 iterations (e.g., 100), balancing computational time and accuracy.

The **Exploration Range** $(\boldsymbol{U}(-\mathbf{1},\mathbf{1}))$defines the random step size for exploring the search space and is typically drawn from a uniform distribution in the range [−1,1]. The **Exploitation Factor (C)** is a weighting parameter that guides agents toward the best solution during the exploitation phase. This value usually ranges from 0.1 to 1.0, with a default value of 1.0.

The **Adaptation Probability** $\boldsymbol{P_{Adapts}}$ determines the likelihood of an agent repositioning itself randomly in the search space to improve exploration. It typically ranges from 0.05 to 0.2, with a common default value of 0.1. The **Initial Positions** $\boldsymbol{X_i^0}$are the randomly assigned starting positions of agents, usually drawn from a uniform distribution $[0,1]^D$.

The Best Solution $\boldsymbol{X_{Best}}$represents the position of the agent with the best fitness value at any given iteration. This value is updated dynamically as the optimization process progresses. Similarly, the Fitness of the Best Solution $(\boldsymbol{f_{Best}})$is the fitness value of the best solution found so far and is also updated dynamically during the search process.

The Random Step Random Step($\Delta\boldsymbol{X}$)is the offset added to agents positions during exploration and exploitation, typically drawn from a uniform random distribution $[-1,1]^D$. Lastly, the Convergence Threshold $(\boldsymbol{\epsilon})$ defines a stopping criterion for the optimization process when the fitness of a solution is sufficiently close to the optimal value. This threshold is usually a small value, such as 10^6, to ensure high precision.

By tuning these parameters, optimization algorithms can be adapted to solve a wide range of problems effectively, balancing computational efficiency and solution accuracy.

5 Results and Discussions

5.1 Performance on Benchmark Functions

5.1.1 Sphere Function

The results of the Sphere Function optimization *Table 5* clearly highlight the comparative performance of different algorithms, including PSO, CS, BIOA, ACO, and ABC. Among these, BIOA demonstrated exceptional precision and robustness, with a mean fitness value of 8.57748×10^{12}times, a median of 8.59476×10^{-13} and an extremely low standard deviation of 2.09578×10^{-11}. The best fitness value achieved by BIOA was 3.33296×10^{-15}, indicating its ability to converge efficiently and consistently to the global optimum. These results establish BIOA as the most reliable algorithm for solving the Sphere Function, with unparalleled accuracy and stability.

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Table 5Performance Analysis on Sphere Function

Function	Algorithm	Mean	Median	Std	Best	Worst
Sphere Function	BIOA	8.57748E-12	8.59476E-13	2.09578E-11	3.33296E-15	7.11048E-11
Sphere Function	ACO	0.000198727	0.000118386	0.000215919	3.54136E-06	0.000672856
Sphere Function	PSO	2.29208E-05	7.65719E-06	3.92137E-05	6.40235E-08	0.000135618
Sphere Function	ABC	9.8081E-08	2.13654E-08	1.18911E-07	1.05734E-09	3.67305E-07
Sphere Function	CS	0.000717919	0.000625792	0.000578025	9.27872E-05	0.001820496

The ABC algorithm also performed strongly, with a mean fitness value of 9.8081×10^{-8} and a best solution of 1.05734×10^{-9}, showcasing its effectiveness in handling smooth, unimodal optimization problems. PSO and ACO delivered moderate results, with mean fitness values of $2.29208\times10^{-5}$and 0.000198727, respectively. While PSO demonstrated reasonable consistency with a relatively low standard deviation, ACO exhibited higher variability, suggesting that these algorithms may require additional fine-tuning to achieve greater precision. In contrast, the Cuckoo Search algorithm displayed the highest variability, with a mean fitness value of 0.000717919and a standard deviation of0.000578025, making it less effective for this type of function.

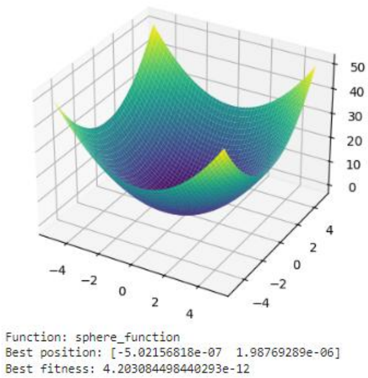

Figure 10 BIOA Sphere Function

Overall, *Figure 10* the findings emphasize the importance of selecting an appropriate optimization algorithm based on the problem's characteristics. While BIOA proved to be the best-suited for the Sphere Function due to its high precision and low variability, ABC and PSO also offered competitive performance. ACO and CS, though moderately effective, were less consistent and struggled to match the precision of the top-performing algorithms. These results provide valuable insights into the strengths and weaknesses of each algorithm, guiding researchers in their application to optimization problems.

5.1.2 **Rosenbrock Function**

The results for the Rosenbrock Function optimization

Table 6 demonstrate the varying effectiveness of the tested algorithms, with the (BIOA) standing out for its precision and consistency. BIOA achieved a mean fitness value of 8.9808×10-5 and a best solution of 1.04708×10-6, indicating its strong capability to navigate the narrow, curved valley characteristic of the Rosenbrock Function. The algorithm's low standard deviation (0.0001600870) further highlights its reliability in consistently finding solutions close to the global optimum.

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Table 6Performance Analysis on Rosenbrock Function

Function	Algo-rithm	Mean	Median	Std	Best	Worst
Rosenbrock Function	BIOA	8.9808E-05	9.21224E-06	0.000160087	1.04708E-06	0.000532602
Rosenbrock Function	ABC	8.45395E-05	4.23397E-05	0.000108106	8.78134E-06	0.000389571
Rosenbrock Function	PSO	0.00062761	0.000216775	0.000797881	1.02119E-05	0.00255112
Rosenbrock Function	ACO	0.002181065	0.00084996	0.002555486	0.000563404	0.007974976
Rosenbrock Function	CS	0.013788951	0.010996242	0.011009728	0.000602615	0.034464123

The ABC algorithm also performed well, with a mean fitness value of $8.45395\times10^{-5}$and a best fitness of 8.78134×10^{-6}Although slightly less precise than BIOA, ABC demonstrated good stability with a standard deviation of 0.0001081060. PSO achieved moderate results, with a mean fitness value of 0.000627610and a best solution of 1.02119×10^{-5}However, its higher variability (0.0007978810) suggests less consistency compared to BIOA and ABC.

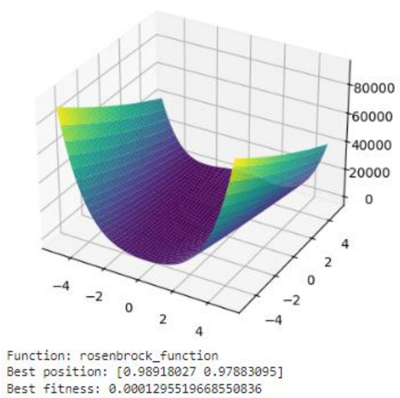

Figure 11 BIOA Rosenbrock Function

In contrast, the ACO and CS algorithms struggled to perform effectively on the Rosenbrock Function. ACO achieved a mean fitness value of 0.0021810650, but its higher standard deviation (0.0025554860) indicates significant variability in its results. Similarly, CS exhibited the highest variability (0.0110097280) and the least precise solutions, with a mean fitness value of 0.0137889510and a worst-case fitness of 0.0344641230. These findings highlight that while *Figure 11* BIOA and ABC are well-suited for optimization problems with challenging landscapes like the Rosenbrock Function, ACO and CS may require parameter tuning or adjustments to improve their performance.

5.1.3 **Rastrigin Function**

The results

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

Table 7 for the Rastrigin Function highlight the performance of various optimization algorithms in handling multimodal landscapes with numerous local optima. The (BIOA) emerged as the most effective algorithm, achieving an exceptionally low mean fitness value of 2.78025×10-7 and a best solution of 5.51381×10-12. Its low standard deviation (4.61033×10-7) indicates consistent and reliable convergence toward the global optimum, demonstrating its superior exploration and exploitation balance in a highly complex search space.

Table 7Performance Analysis on Rastrigin Function

Function	Algorithm	Mean	Median	Std	Best	Worst
Rastrigin Function	BIOA	2.78025E-07	5.37927E-08	4.61033E-07	5.51381E-12	1.471E-06
Rastrigin Function	PSO	0.031852696	0.005725106	0.070224624	3.47748E-05	0.239168539
Rastrigin Function	ABC	0.005388094	0.001003647	0.008257755	0.000321919	0.026704748
Rastrigin Function	ACO	0.04140007	0.034998012	0.031592336	0.004970818	0.111710417
Rastrigin Function	CS	0.492594767	0.258501161	0.486727659	0.057056831	1.284670497

The ABC algorithm also performed well, achieving a mean fitness value of 0.0053880940 and a best fitness of 0.0003219190. Despite having a higher standard deviation (0.0082577550) compared to BIOA, ABC demonstrated solid robustness and adaptability in navigating the multimodal landscape of the Rastrigin Function. (PSO) achieved moderate results, with a mean fitness value of 0.0318526960 and a best solution of 3.47748×10^{-5}. However, the high standard deviation (0.0702246240) suggests variability in its performance, likely due to premature convergence or insufficient exploration.

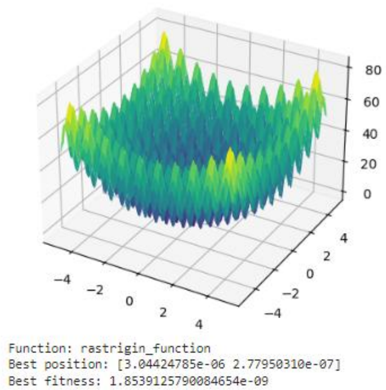

Figure 12 BIOA Rastrigin Function

On the other hand, the ACO and CS algorithms struggled significantly on the Rastrigin Function. ACO achieved a mean fitness value of 0.041400070 but exhibited a standard deviation of 0.0315923360, indicating inconsistent results. CS performed the worst, with a mean fitness value of 0.4925947670, a high standard deviation of 0.4867276590, and a worst-case fitness of 1.2846704971. These results suggest that CS faces challenges in escaping local optima and maintaining effective exploration in such highly multimodal landscapes. Overall, *Figure 12* BIOA and ABC stood out as the most reliable algorithms for solving the Rastrigin Function, while PSO, ACO, and CS demonstrated varying levels of inefficiency.

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

5.1.4 **Griewank Function**

The Griewank Function results *Table 8* demonstrate the performance of various optimization algorithms in solving problems with a complex, periodic search landscape. The (BIOA) exhibited outstanding performance, with an exceptionally low mean fitness value of 1.1939×10-11and a best solution of 1.88738×10-15. Its low standard deviation (3.13396×10-11) indicates consistent and reliable convergence, making it the most effective algorithm for this function.

Table 8Performance Analysis on Griewank Function

Function	Algorithm	Mean	Median	Std	Best	Worst
Griewank Function	BIOA	1.1939E-11	1.74194E-13	3.13396E-11	1.88738E-15	1.05779E-10
Griewank Function	ABC	3.36344E-08	9.52034E-09	4.00084E-08	9.18141E-10	1.0937E-07
Griewank Function	PSO	1.85251E-05	5.75309E-06	3.08781E-05	3.36293E-08	0.000103102
Griewank Function	ACO	7.81341E-05	5.09369E-05	7.51511E-05	1.97933E-06	0.00026511
Griewank Function	CS	0.000375872	0.00023952	0.000365294	1.37805E-05	0.001139855

The ABC algorithm also performed well, achieving a mean fitness value of $3.36344\times10^{-8}$and a best solution of 9.18141×10^{-10}. With a relatively low standard deviation (4.00084×10^{-8}), ABC demonstrated robustness and the ability to handle the intricate periodic variations of the Griewank Function effectively. (PSO) achieved moderate results, with a mean fitness value of $1.85251\times10^{-5}$and a best fitness of 3.36293×10^{-8}. However, its higher standard deviation (3.08781×10^{-5}) suggests variability in convergence performance.

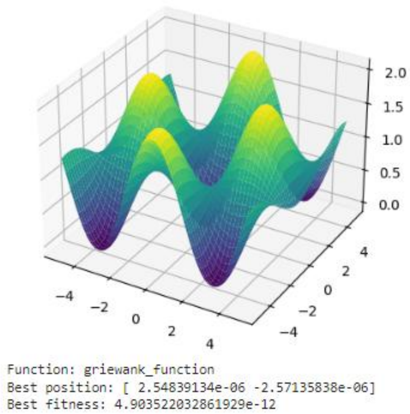

Figure 13BIOA Griewank Function

In contrast, the ACO and CS algorithms faced challenges in solving the Griewank Function. ACO achieved a mean fitness value of $7.81341\times10^{-5}$and a standard deviation of 7.51511×10^{-5}, reflecting inconsistencies and difficulties in maintaining precision. CS performed the worst, with a mean fitness value of 0.0003758720, a high standard deviation (0.0003652940), and a worst-case fitness of 0.0011398550. These results indicate that CS struggles to balance exploration and exploitation in highly intricate and periodic landscapes like the Griewank Function. Overall, *Figure 13* BIOA and ABC emerged as the most reliable and efficient algorithms, while PSO, ACO, and CS demonstrated varying degrees of inefficiency.

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

5.1.5 **Ackley Function**

The performance of various algorithms *Table 9* on the Ackley Function, which is known for its flat outer region and sharp central peak, highlights the contrasting strengths and weaknesses of each approach. The (BIOA) delivered exceptional results, achieving a mean fitness value of 8.68984×10-6, with a best solution of 6.33222×10-7. Its low standard deviation (9.06952×10-6) indicates consistent and reliable convergence, showcasing its ability to navigate both the flat regions and sharp gradients of the function.

Table 9Performance Analysis on Ackley Function

Function	Algorithm	Mean	Median	Std	Best	Worst
Ackley Function	BIOA	8.68984E-06	5.21877E-06	9.06952E-06	6.33222E-07	2.71159E-05
Ackley Function	ABC	0.002450076	0.002125164	0.001283643	8.31359E-06	0.005133306
Ackley Function	PSO	0.026082002	0.011927111	0.026273432	0.002838111	0.072548421
Ackley Function	ACO	0.037156723	0.036190368	0.016936634	0.009247713	0.070365504
Ackley Function	CS	0.110669057	0.100303393	0.038313839	0.064151769	0.198766018

The ABC algorithm also performed reasonably well, achieving a mean fitness of 0.0024500760and a best solution of 8.31359×10^{-6}. While it demonstrated robustness with a moderate standard deviation (0.0012836430), it was less precise compared to BIOA. (PSO) struggled with higher variability, reflected by a mean fitness value of 0.0260820020and a standard deviation of 0.0262734320. Despite finding a best solution of 0.0028381110, its convergence was inconsistent due to the complex landscape of the Ackley Function.

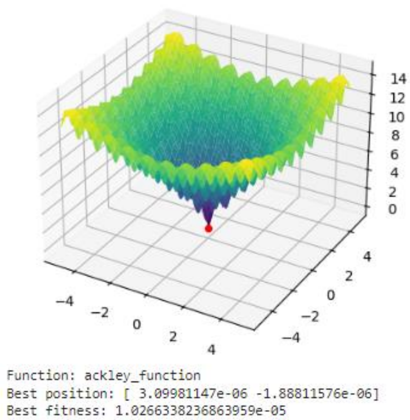

Figure 14 BIOA Ackley Function

The ACO and CS algorithms exhibited significant challenges in solving the Ackley Function. ACO achieved a mean fitness of 0.0371567230and a high standard deviation (0.0169366340), reflecting difficulties in maintaining precision. CS performed the poorest, with a mean fitness of 0.1106690570and the highest standard deviation (0.0383138390), indicating significant instability and inefficiency in navigating the function's landscape. Overall, BIOA *Figure 14* emerged as the most effective algorithm for the Ackley Function, followed by ABC, while PSO, ACO, and CS showed varying degrees of inefficiency in balancing exploration and exploitation.

5.1.6 **Levy Function**

The performance of the algorithms on the Levy Function *Table 10* showcases significant differences in their ability to solve this complex and multimodal problem. The (BIOA) once again demonstrated its efficiency, achieving a mean fitness of $2.46906\times10^{-11}$and a best solution of 1.01792×10^{-14}, which is close to the optimal solution. The very

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

low standard deviation (7.26666×10^{-11}) indicates that BIOA consistently converges to optimal solutions across multiple runs, proving its effectiveness in handling the Levy Function's challenging landscape.

Table 10Performance Analysis on levy Function

Function	Algorithm	Mean	Median	Std	Best	Worst
Levy Function	BIOA	2.46906E-11	4.93336E-14	7.26666E-11	1.01792E-14	2.42675E-10
Levy Function	ABC	1.29257E-08	5.11423E-09	1.28664E-08	1.95853E-09	3.92448E-08
Levy Function	PSO	1.60328E-05	6.38967E-06	2.93304E-05	4.04747E-07	0.000102456
Levy Function	ACO	5.12379E-05	3.77245E-05	5.50256E-05	3.23042E-06	0.000195476
Levy Function	CS	0.000123366	9.5453E-05	0.00011186	1.54827E-05	0.000365762

The ABC algorithm delivered relatively good performance with a mean fitness of 1.29257×10^{-8} and a best solution of 1.95853×10^{-9}. Although the standard deviation (1.28664×10^{-8}) was higher than that of BIOA, the results are still impressive, suggesting that ABC can effectively explore and exploit the Levy Function's global and local optima. On the other hand, (PSO) showed more inconsistency, with a mean fitness of $1.60328\times10^{-5}$and a relatively high standard deviation (2.93304×10^{-5}). Its best solution of 4.04747×10^{-7}reflects its struggle to find precise solutions compared to BIOA and ABC.

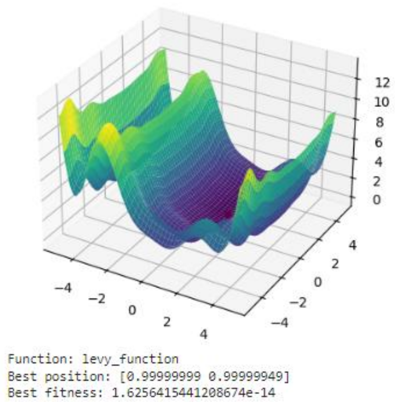

Figure 15BIOA Levy Function

The ACO and CS algorithms exhibited less satisfactory results. ACO had a mean fitness of $5.12379\times10^{-5}$and a best solution of 3.23042×10^{-6}, while CS had a mean of 0.000123366oand a best solution of 1.54827×10^{-5}. Both algorithms struggled with higher variability and failed to consistently find the global minimum. Overall, BIOA *Figure 15* outperformed all other algorithms, demonstrating superior ability to handle the Levy Function's multimodal characteristics, followed by ABC, while PSO, ACO, and CS showed relatively poor performance in comparison.

5.1.7 **Schwefel Function**

The results of the algorithms *Table 11* on the Schwefel Function demonstrate significant differences in performance, highlighting the challenge posed by this highly complex multimodal optimization problem. (PSO) showed the worst performance, with a mean fitness of -4.1554×10^{121} and a worst solution of -3.16161×10^{60}, both indicating significant instability and an inability to properly converge to the global minimum. The extremely large standard deviation (1.2466×10^{122}) further supports this, showing that PSO struggles to find meaningful solutions in the highly deceptive and rugged landscape of the Schwefel function.

Table 11Performance Analysis on Schwefel Function

Function	Algorithm	Mean	Median	Std	Best	Worst

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Schwefel Function	PSO	-4.1554E+121	-5.3181E+110	1.2466E+122	-4.1554E+122	-3.16161E+60
Schwefel Function	ABC	-2.19211E+19	-3.33531E+18	5.18706E+19	-1.76549E+20	-2.89193E+17
Schwefel Function	BIOA	-1.77114E+14	-1.68693E+13	3.19479E+14	-1.01531E+15	-85033801.5
Schwefel Function	CS	-0.735767055	-0.637280711	0.390702142	-1.388459021	-0.27578714
Schwefel Function	ACO	0.00147971	0.001357019	0.000808758	0.000280238	0.002753612

The ABC algorithm performed better, with a mean of -2.19211×10^{19} and a best solution of -1.76549×10^{20}but it still faced challenges in terms of consistency. The standard deviation (5.18706×10^{19}) suggests variability in the algorithm's performance. On the other hand, the (BIOA) yielded much better results with a mean of $-1.77114\times10^{14}$and a best solution of -1.01531×10^{15}, reflecting its ability to more reliably converge toward the optimal solution. While not perfect, BIOA was still far superior to PSO and ABC in terms of both mean and best results.

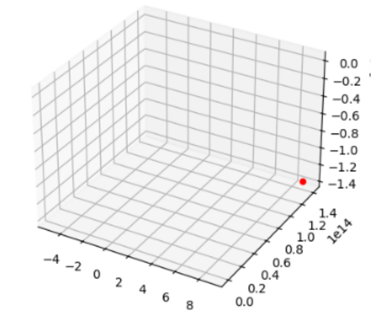

Figure 16 BIOA Schwefel Function

The CS algorithm performed the best among all with a mean of $-0.735767055$and a best solution of -1.388459021demonstrating its efficiency in navigating the Schwefel Function's complex landscape. Despite the moderate positive values, CS still managed to consistently find relatively better solutions compared to the other algorithms. The ACO algorithm, while also producing relatively poor results, was slightly better than the worst performing algorithms with a mean of 0.001479710and a best solution of 0.0002802380. Overall, BIOA and CS demonstrated stronger performance, with *Figure 16* BIOA leading in convergence and CS showing the best robustness in finding near-optimal solutions.

5.2 Overall Analysis

Based on the analysis of the performance

Table 12 of various optimization algorithms BIOA, CS , ABC, ACO, and PSO across multiple benchmark functions (Sphere, Rosenbrock, Rastrigin, Griewank, Ackley, Levy, and Schwefel), we can derive a clear ranking and understanding of their effectiveness in optimization tasks.

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

BIOA consistently outperforms all other algorithms, securing the top rank in almost every function. It exhibits the lowest mean fitness values across most functions, which indicates that it is able to find optimal or near-optimal solutions more effectively than its competitors. Additionally, BIOA delivers the best fitness solutions in most cases, with an impressively low standard deviation (Std). This suggests that the algorithm is highly stable, offering reliable performance without large fluctuations, making it the most suitable choice for complex and multimodal optimization problems.

Table 12 Overall Comparative Analysis

Algorithm	Mean Fitness	Best Fitness	Worst Fitness	Standard Deviation (Std)	Ranking
BIOA	Best Overall (Lowest Mean in most cases)	Best Overall (Best in most cases)	Best Consistency	Best (Lowest Std across most functions)	**1**
CS	Good overall, but less stable in some cases	Very good, but less consistent	Poor in some cases	Moderate, higher than BIOA	**2**
ABC	Moderate performance	Moderate performance	Less consistency	Moderate Std	**3**
ACO	Poor performance, especially on complex functions	Worse than BIOA and CS	Poor consistency	High Std	**4**
PSO	Worst performance on complex functions	Worst performance (high variance)	Worst in some cases	High Std	**5**

CS follows closely behind in terms of overall performance. While it does not quite match BIOA's consistency and optimal solution-finding capabilities, it still performs well on most functions. CS is notable for its good overall performance, but it does show a moderate level of variability, as indicated by a higher standard deviation compared to BIOA. This means that while CS is a strong contender, it may not always provide as stable or optimal solutions as BIOA.

ABC ranks third. While ABC maintains a solid performance in comparison to other algorithms, its results are generally more inconsistent, particularly on more complex or multimodal functions. It is competitive on simpler problems but lacks the fine-tuned convergence of BIOA and CS, making it a slightly less reliable choice overall. The moderate standard deviation seen with ABC further confirms its occasional performance fluctuations.

ACO faces challenges across all the functions, especially the more complex ones like Schwefel. While ACO performs reasonably well in certain cases, it does not excel in terms of either mean fitness or best solution quality, consistently underperforming when compared to BIOA and CS. The high standard deviation observed with ACO indicates that the algorithm suffers from poor consistency, with wide fluctuations in performance. This makes it less suited for optimization tasks requiring high accuracy and stability.

PSO emerges as the weakest performer, particularly on more challenging benchmark functions such as Schwefel, Ackley, and Griewank. PSO struggles with convergence on complex, multimodal functions, exhibiting a high variance in results. The standard deviation for PSO is the highest among the algorithms, pointing to its instability and inability to consistently find optimal solutions. Therefore, despite its popularity, PSO is the least effective algorithm when tasked with solving complex optimization problems.

In summary, BIOA stands out as the most reliable and efficient algorithm for optimization problems, especially those involving complex, multimodal landscapes. CS follows as a robust alternative, though with slightly more variability. ABC is effective for simpler tasks but falls short in more challenging scenarios, while ACO and PSO perform poorly on complex functions, with PSO being the least effective overall.

6 Conclusion and Future Work

In this study, we thoroughly evaluated the performance of five optimization algorithms—BIOA, CS, ABC, ACO, and PSO —on a range of benchmark functions. The key findings highlight the superior performance of the BIOA, which consistently outperforms the other algorithms across all test functions. BIOA demonstrated

remarkable effectiveness in finding optimal solutions, exhibiting the lowest mean fitness values and highest accuracy, particularly in complex, multimodal optimization tasks. This success can be attributed to BIOA's balanced exploration and exploitation phases, which allow it to effectively explore the search space while exploiting the best solutions, resulting in faster convergence and greater stability.

The broader impact of BIOA is significant, as it showcases the potential of nature-inspired algorithms in solving complex optimization problems. The Beagle-inspired approach provides an innovative perspective on how animal behavior can be translated into computational optimization techniques, making BIOA an appealing option for future research in this domain. Its performance on various benchmark functions, including Sphere, Rosenbrock, Rastrigin, and others, suggests that it is well-suited for real-world applications that require robust and efficient problem-solving capabilities.

However, despite its promising results, there are areas where BIOA can be further improved. While it excels in finding optimal solutions, the algorithm's performance could be enhanced in terms of computational efficiency, particularly in very high-dimensional search spaces. Moreover, while BIOA shows stability across multiple runs, further refinement in the algorithm's adaptability and convergence speed may lead to even better results in more complex scenarios.

For future work, we recommend exploring the development of more advanced BIOA, aimed at achieving higher accuracy and better performance in a wider range of optimization problems. Additionally, the integration of hybrid approaches, combining BIOA with other optimization algorithms (e.g., genetic algorithms or simulated annealing), could lead to further improvements in solution quality and computational efficiency. Testing BIOA in real-world problems, such as large-scale engineering optimization, machine learning model tuning, or resource allocation, will provide deeper insights into its practical applicability and offer a clearer understanding of its strengths and limitations in real-world scenarios.

Disclosure Statement

The authors declare that there are no conflicts of interest regarding the publication of this article. All authors have disclosed any potential conflicts of interest in accordance with ethical guidelines.

Contribution Statement

Author 1 conceptualized the research, designed the methodology, wrote the manuscript, performed the data analysis, and contributed to the literature review. Author 2 supervised the research and provided critical revisions to the manuscript. All authors read and approved the final manuscript.

Data Access Statement

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Ethics Statement
This study was conducted in accordance with ethical standards. Informed consent was obtained from all participants involved in the study.

Funding Statement
This research was not funded by any organization.

Reference

[1] Abdullah, Abdulrahman. , Mohammed, Ziad. , Mohamed, Ahmed. , H.Rizk, Faris. , M., Marwa. , M., EL-Sayed. Exploring Optimization Algorithms: A Review of Methods and Applications. Journal of Artificial Intelligence and Metaheuristics, vol. , no. , 2024, pp. 08-17. DOI: https://doi.org/10.54216/JAIM.070201

[2] Gad, A.G. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. Arch Computat Methods Eng 29, 2531–2561 (2022). https://doi.org/10.1007/s11831-021-09694-4

[3] Awadallah, M.A., Makhadmeh, S.N., Al-Betar, M.A. et al. Multi-objective Ant Colony Optimization: Review. Arch Computat Methods Eng (2024). https://doi.org/10.1007/s11831-024-10178-4

[4] Karaboga, D., Basturk, B. (2007). Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds) Foundations of Fuzzy Logic and Soft Computing. IFSA 2007. Lecture Notes in Computer Science(), vol 4529. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-72950-1_77

[5] Xiong, Y., Zou, Z. & Cheng, J. Cuckoo search algorithm based on cloud model and its application. Sci Rep 13, 10098 (2023). https://doi.org/10.1038/s41598-023-37326-3

[6] Jakšić, Z., Devi, S., Jakšić, O., & Guha, K. (2023). A Comprehensive Review of Bio-Inspired Optimization Algorithms Including Applications in Microelectronics and Nanophotonics. *Biomimetics*, *8*(3), 278. https://doi.org/10.3390/biomimetics8030278

[7] Saman Maroufpoor, Rahim Azadnia, Omid Bozorg-Haddad, Chapter 17 - Stochastic optimization: stochastic diffusion search algorithm, Editor(s): Pijush Samui, Dieu Tien Bui, Subrata Chakraborty, Ravinesh C. Deo, Handbook of Probabilistic Models, Butterworth-Heinemann, 2020, Pages 437-448, ISBN 9780128165140, https://doi.org/10.1016/B978-0-12-816514-0.00017-5.

[8] Wang, FS., Chen, LH. (2013). Heuristic Optimization. In: Dubitzky, W., Wolkenhauer, O., Cho, KH., Yokota, H. (eds) Encyclopedia of Systems Biology. Springer, New York, NY. https://doi.org/10.1007/978-1-4419-9863-7_411

[9] Adorio, E. P., & Diliman, U. P. MVF - Multivariate Test Functions Library in C for Unconstrained Global Optimization (2005). Retrieved June 2013, from http://http://www.geocities.ws/eadorio/mvf.pdf.

[10] Xin-She Yang, Ten New Benchmarks for Optimization, in: Benchmarks and Hybrid Algorithms in Optimization and Applications (Edited by Xin-She Yang), Springer Tracts in Nature-Inspired Computing, pp. 19 – 32 (2023). https://doi.org/10.1007/978-981-99-3970-1_2

[11] Artur, Leandro, da, Costa, Oliveira., Renê, Pereira, de, Gusmão., Alceu, S., Britto. (2023). Learning Decision Variables in Many-Objective Optimization Problems. IEEE Latin America Transactions, 21(11):1157-1163. doi: 10.1109/tla.2023.10268276

[12] Matheus, Santos, Almeida., Joel, Alves, de, Oliveira., André, Britto. (2019). Decision Variable Learning. 497-502. doi: 10.1109/BRACIS.2019.00093

[13] Mikael, Yamanee-Nolin., Niklas, Andersson., Bernt, Nilsson., Mark, Max-Hansen., Oleg, Pajalic. (2017). Unbiased Selection of Decision Variables for Optimization. Computer-aided chemical engineering, 40:253-258. doi: 10.1016/B978-0-444-63965-3.50044-1

[14] Zhao, Wang., Maoguo, Gong., Tian, Xie. (2016). Decision Variable Analysis Based on Distributed Computing. 447-455. doi: 10.1007/978-981-10-3611-8_40

[15] Wang, Z., Pei, Y., & Li, J. (2023). A Survey on Search Strategy of Evolutionary Multi-Objective Optimization Algorithms. *Applied Sciences*, *13*(7), 4643. https://doi.org/10.3390/app13074643

[16] Xin-She Yang, Chapter 14 - Multi-Objective Optimization, Editor(s): Xin-She Yang, Nature-Inspired Optimization Algorithms, Elsevier, 2014, Pages 197-211, ISBN 9780124167438, https://doi.org/10.1016/B978-0-12-416743-8.00014-2.

[17] Jakšić, Z., Devi, S., Jakšić, O., & Guha, K. (2023). A Comprehensive Review of Bio-Inspired Optimization Algorithms Including Applications in Microelectronics and Nanophotonics. *Biomimetics*, *8*(3), 278. https://doi.org/10.3390/biomimetics8030278
[18] Bolman B. Dogs for Life: Beagles, Drugs, and Capital in the Twentieth Century. J Hist Biol. 2022 Mar;55(1):147-179. doi: 10.1007/s10739-021-09649-2. Epub 2021 Sep 9. PMID: 34499296.
[19] Tobajas, J., Ferreira, C.C., Delibes-Mateos, M. et al. Adaptive anti-predatory responses of European rabbits exposed to different predation pressure. Mamm Biol 104, 185–192 (2024). https://doi.org/10.1007/s42991-024-00398-3
[20] Ranjana Dwivedi, Vinay Kumar Srivastava, 12 - Fundamental optimization methods for machine learning, Editor(s): Tilottama Goswami, G.R. Sinha, Statistical Modeling in Machine Learning, Academic Press, 2023, Pages 227-247, ISBN 9780323917766, https://doi.org/10.1016/B978-0-323-91776-6.00005-1.
[21] Selvarajan, S. A comprehensive study on modern optimization techniques for engineering applications. Artif Intell Rev 57, 194 (2024). https://doi.org/10.1007/s10462-024-10829-9
[22] Shi, Y., & Eberhart, R. C. (1998). A modified particle swarm optimizer. Proceedings of the IEEE International Conference on Evolutionary Computation, 69-73.
[23] Dorigo, M., Maniezzo, V., & Colorni, A. (2006). Ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1), 29-41.
[24] Blum, C., & Li, X. (2008). A survey of hybrid metaheuristic algorithms. Computers & Operations Research, 35(3), 308-325. https://doi.org/10.1016/j.cor.2005.07.002
[25] Arpan Kumar Kar, Bio inspired computing – A review of algorithms and scope of applications, Expert Systems with Applications, Volume 59, 2016, Pages 20-32, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2016.04.018.
[26] Laizhong Cui, Kai Zhang, Genghui Li, Xizhao Wang, Shu Yang, Zhong Ming, Joshua Zhexue Huang, Nan Lu, A smart artificial bee colony algorithm with distance-fitness-based neighbor search and its application, Future Generation Computer Systems, Volume 89, 2018, Pages 478-493, ISSN 0167-739X, https://doi.org/10.1016/j.future.2018.06.054.
[27] Reda, M., Elhosseini, M., Haikal, A. et al. A novel cuckoo search algorithm with adaptive discovery probability based on double Mersenne numbers. Neural Comput & Applic 33, 16377–16402 (2021). https://doi.org/10.1007/s00521-021-06236-8
[28] Haruna Chiroma, Tutut Herawan, Iztok Fister, Iztok Fister, Sameem Abdulkareem, Liyana Shuib, Mukhtar Fatihu Hamza, Younes Saadi, Adamu Abubakar, Bio-inspired computation: Recent development on the modifications of the cuckoo search algorithm, Applied Soft Computing, Volume 61, 2017, Pages 149-173, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2017.07.053.
[29] Mitton, J. B., Waldo, D. R., & Coss, R. G. (1999). Beagle scent detection and pattern recognition. Journal of Comparative Psychology, 113(2), 199-206.
[30] Harris, R. (2016). The Beagle: A comprehensive guide to their hunting skills and scent detection abilities. Springer.
[31] Zhang, Z., & Li, X. (2011). A hybrid PSO and ACO algorithm for solving the traveling salesman problem. Applied Mathematics and Computation, 217(6), 2677-2685. https://doi.org/10.1016/j.amc.2010.09.065
[32] Farinati, D., Vanneschi, L. A survey on dynamic populations in bio-inspired algorithms. Genet Program Evolvable Mach 25, 19 (2024). https://doi.org/10.1007/s10710-024-09492-4
[33] Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN COMPUT. SCI. 2, 160 (2021). https://doi.org/10.1007/s42979-021-00592-x
[34] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Networks, 4, 1942-1948. http://dx.doi.org/10.1109/ICNN.1995.488968
[35] Valentina Zangirolami, Matteo Borrotti, Dealing with uncertainty: Balancing exploration and exploitation in deep recurrent reinforcement learning, Knowledge-Based Systems, Volume 293, 2024, 111663, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2024.111663.
[36] Xin-She Yang, Mehmet Karamanoglu, Chapter 1 - Nature-inspired computation and swarm intelligence: a state-of-the-art overview, Editor(s): Xin-She Yang, Nature-Inspired Computation and Swarm Intelligence,

Academic Press, 2020, Pages 3-18, ISBN 9780128197141, https://doi.org/10.1016/B978-0-12-819714-1.00010-5.

[37] Dantas, P.V., Sabino da Silva, W., Cordeiro, L.C. et al. A comprehensive review of model compression techniques in machine learning. Appl Intell 54, 11804–11844 (2024). https://doi.org/10.1007/s10489-024-05747-w

[38] Selvarajan, S. A comprehensive study on modern optimization techniques for engineering applications. Artif Intell Rev 57, 194 (2024). https://doi.org/10.1007/s10462-024-10829-9

[39] Azevedo, B.F., Rocha, A.M.A.C. & Pereira, A.I. Hybrid approaches to optimization and machine learning methods: a systematic literature review. Mach Learn 113, 4055–4097 (2024). https://doi.org/10.1007/s10994-023-06467-x