# Extended Honey Badger Algorithm: A Competitive Metaheuristic for Solving Global Optimization Problems

Samindar J. Vibhute[1], Chetan S. Arage[2], Vishal S. Pawar[3], Abhishek A. Patil[4]
[1]*Department Computer Science & Engineering, Sanjay Ghodawat University, Kolhapur, Maharashtra, India*
[2]*Department Computer Science & Engineering, Sanjay Ghodawat University, Kolhapur, Maharashtra, India*
[3]*Department Computer Science & Engineering, Ashokrao Mane Group of Institutions, Kolhapur, India*
[4]*Department Computer Science & Engineering, Tatyasaheb Kore Institute of Engineering & Technology, Kolhapur, India*

***Abstract— This study proposes the Extended Honey Badger Optimization (EHBO) algorithm, an enhancement of the standard Honey Badger Algorithm (HBA), inspired by the unique predatory behavior of honey badgers. The EHBO introduces adaptive mechanisms designed to balance exploration and exploitation capabilities more effectively. To assess its efficacy, the EHBO is benchmarked against widely-used optimization algorithms, namely Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and the base HBA across standard benchmark functions such as Rastrigin, Rosenbrock, and Sphere functions. Experimental results demonstrate that EHBO outperforms ACO consistently and performs competitively against HBA and PSO, especially on unimodal landscapes like the Sphere function. However, PSO retains superiority across all test cases, particularly on complex multimodal problems. The findings suggest that while EHBO improves upon its base algorithm, further refinement is necessary to rival the performance of mature swarm intelligence algorithms like PSO. The results underscore EHBO's potential for further development as a robust optimization tool for continuous optimization problems.***

***Keywords— Extended Honey Badger Optimization (EHBO), Swarm Intelligence, Benchmark Functions, Global Optimization***

## I. INTRODUCTION

Swarm intelligence optimization is a rapidly evolving field in computational intelligence that draws inspiration from the collective behaviour of social organisms in nature (Parpinelli & Lopes, 2011). This approach to problem-solving and optimization is based on the observation that simple, decentralized agents can exhibit complex and intelligent behaviour when working together as a group. Swarm intelligence algorithms mimic the decision-making processes and movement patterns of various biological systems, such as ant colonies, bird flocks, and fish schools, to solve complex optimization problems in diverse domains. The concept of swarm intelligence was first introduced by Beni and Wang (1989) in the context of cellular robotic systems. Since then, it has gained significant attention from researchers and practitioners across multiple disciplines, including computer science, engineering, and operations. The appeal of swarm intelligence lies in its ability to tackle complex, high-dimensional problems with relatively simple algorithmic structures and minimal computational requirements. Swarm intelligence optimization algorithms typically operate by iteratively improving a population of candidate solutions through local interactions and information sharing among individual agents. These algorithms are characterized by their self-organization, adaptability, and robustness in the face of changing environments and problem constraints. Some of the most popular swarm intelligence optimization techniques include Particle Swarm Optimization (PSO) (Jabeen et al., 2009), Ant Colony Optimization (ACO) (Ünal et al., 2013) , and Artificial e colony (ABC) algorithms (Yi & He, 2014).The applications of swarm intelligence optimization span a wide range of fields, including, but not limited to, engineering design and optimization(Martins & Ning, 2021), resource allocation and scheduling (Guo & Liu, 2019) , Data mining a clustering,  network routing and load balancing, image and signal processing, robotics, and autonomous systems. As the complexity of real-world optimization problems continues to grow, swarm intelligence optimization techniques offer promising solutions owing to their scalability, parallelizability, and ability to handle multi-objective optimization scenarios. This introduction provides an overview of swarm intelligence optimization, its fundamental principles, and its significance in addressing contemporary challenges in various domains.

Research gaps in nature-inspired algorithms Theoretical foundations: Lack of comprehensive mathematical models to explain the convergence and behaviour of swarm intelligence algorithms in complex optimization landscapes.(Schranz & Sende, 2020) Parameter tuning: Insufficient research on adaptive parameter tuning mechanisms to optimize algorithm performance across diverse problem domains.(Smit & Eiben, 2009) Scalability: Limited understanding of how swarm intelligence algorithms perform on large-scale, high-dimensional problems and in distributed computing environments.(Schranz & Sende, 2020). Constraint handling: Inadequate exploration of efficient constraint-handling techniques for swarm intelligence algorithms in highly constrained optimization problems. (Asif Jan et al., 2021). Dynamic environments: Need for more robust swarm intelligence algorithms capable of adapting to rapidly changing optimization landscapes and objectives. (Qu et al., 2019). Multi-objective optimization: Further research is required to balance exploration and exploitation in multi-objective swarm intelligence algorithms. ("Balancing Exploration and Exploitation with Decomposition-Based Dynamic Multi-Objective Evolutionary Algorithm," 2021). Hybridization strategies: Lack of systematic approaches for combining swarm intelligence algorithms with other optimization techniques or machine-learning methods. (Poposki, 2022) Real-time applications: Insufficient multi objective of swarm intelligence algorithms in real-time decision-making scenarios with strict time constraints. Research scope of the hybrid approaches and extended versions1. Meta-heuristic hybridization: Combining multiple swarm intelligence algorithms or integrating them with other meta-heuristics to leverage complementary strengths.(Davis & Papageorgiou, 2021) Machine learning integration: Incorporating machine learning techniques to enhance the learning capabilities and adaptability of swarm intelligence algorithms.(Silahtaroğlu, 2024) Memetic algorithms: Developing hybrid algorithms that combine population-based search with local search methods to improve solution quality.(Tenne, 2012) Quantum-inspired extensions: exploring quantum computing principles to enhance the exploration and exploitation capabilities of existing swarm intelligence algorithms. Multi-swarm approaches: Investigating the potential of multiple interacting swarms to solve complex multimodal optimization problems more effectively.

While the honey badger's physical attributes, such as its stocky build and distinctive black and white fur, make it easily recognizable, its behavioural traits, including aggression, intelligence, and adaptability, set it apart from other small carnivores and inspire the design of robust optimization algorithms. The honey badger's distinctive physical characteristics, including its robust physique and characteristic black and white pelages, render it readily identifiable. However, it is the species' behavioural attributes—notably its aggressive nature, cognitive capabilities, and adaptability—that distinguish it from other small carnivores and serve as inspiration for the development of resilient optimization algorithms. These behavioural traits have led to the honey badger being considered one of the most fearless and tenacious animals in the wild. Its ability to tackle challenges and overcome obstacles has made it a symbol of resilience. This reputation has sparked interest among researchers and engineers seeking to emulate the problem-solving skills of honey badgers in artificial intelligence and optimization techniques.

Image: Honey Badger

## II. LITERATURE REVIEW

The potential scope for developing or expanding an optimization algorithm inspired by the honey badger's characteristics could encompass the following aspects: aggressive exploration, developing mechanisms to thoroughly search the solution space, emulating the persistent nature of the honey badger. Implement adaptive strategies that enable the algorithm to modify its search parameters based on the problem landscape, reflecting the honey badger's adaptability (Frey, 2023). Resilience is a local artificial feature that allows the algorithm to overcome local optima, inspired by the honey badger's ability to surmount obstacles. Multi-objective optimization: Incorporate the capability to manage multiple objectives concurrently, mirroring the honey badger's diverse problem-solving abilities (Frey, 2024). Cognitive learning: Implement machine learning techniques to enhance the algorithm's performance over time (Huang et al., 2025), inspired by the cognitive capabilities of honey badgers. Robustness to environmental changes: Develop mechanisms to maintain performance in dynamic or noisy environments, reflecting the resilience of honey badgers (Frey, 2023). Explore parallel implementation to improve computational efficiency, inspired by the honey badger's ability to multitask. Investigate the potential of integrating the honey badger-inspired algorithm with other optimization techniques to create more sophisticated hybrid algorithms. Develop self-adaptive parameter tuning mechanisms to optimize the algorithm's performance across various problem domains. (Li et al., 2022) Benchmark testing: Conduct comprehensive benchmark tests to evaluate and compare the performance of the algorithm against existing optimization methods. (Atila et al., 2019)

Swarm intelligence optimization algorithms are computational methods inspired by the collective behaviour of social organisms in nature. These algorithms mimic the decision-making processes and movement patterns of biological systems such as ant colonies, bird flocks, and fish schools to solve complex optimization problems. Swarm intelligence techniques typically operate by iteratively improving a population of candidate solutions through local interactions and information-sharing among individual agents. Key characteristics include self-organization, adaptability, and robustness in changing environments. Population intelligence includes the Particle Swarm Optimization (PSO) (Kumar et al., 2015), Ant Colony Optimization (ACO) (Darius et al., 2022), and Artificial Bee Colony (ABC) algorithms (Liang et al., 2020). Swarm intelligence methods have found applications in diverse fields such as engineering design, resource allocation, data mining, network routing, and robotics. Their ability to handle high-dimensional problems with relatively simple structures and minimal computational requirements makes them attractive for tackling increasingly complex real-world optimization challenges (Mohanty, 2018).

Nature-inspired algorithms are a class of computational methods (Yang, 2017) that draw inspiration from biological systems, natural phenomena, and evolutionary processes to solve complex optimization problems. These algorithms mimic the behaviours, strategies, and mechanisms observed in nature in order to efficiently search for optimal solutions in various domains. Examples include genetic algorithms inspired by natural selection, particle swarm optimization based on bird flocking behaviour (Jung et al., 2015), and ant colony optimization derived from the foraging patterns of ants (Popescu, 2023). Nature-inspired algorithms are characterized by their ability to handle high-dimensional, non-linear problems, adapt to changing environments, and find near-optimal solutions with relatively low computational complexity. They have been successfully applied to diverse fields, such as engineering design, resource allocation, scheduling, data mining, and machine learning, demonstrating their versatility and effectiveness in tackling real-world optimization challenges.

Explore hybrid approaches that combine the strengths of multiple optimization techniques to enhance the algorithm's versatility and efficiency (Walsh, 2002). Investigate the scalability and computational complexity of the algorithm to ensure its effectiveness in large-scale optimization problems. Nature inspired processing capabilities to leverage multicore architectures and non linier execution time of the algorithm.

Over the years, numerous metaheuristic algorithms have been proposed to solve complex optimization problems, with nature-inspired algorithms gaining significant traction. Among these, the Honey Badger Algorithm (HBA) has demonstrated promising results in global optimization tasks due to its unique hunting and digging behaviours modelled after real-life honey badgers (Hashim et al., 2022). However, despite its effectiveness, the standard HBA has certain limitations, such as a tendency to get trapped in local optima and relatively slow convergence rates. These limitations have motivated researchers to enhance the baseline HBA for better performance in various optimization tasks.

One notable improvement was proposed by Rezaee et al. (2023), who introduced an Enhanced Honey Badger Algorithm (EHBA) aimed at strengthening HBA's exploration and exploitation balance. Their work incorporated a modified density factor, adaptive weight schemes, and neighbourhood search mechanisms to avoid premature convergence. EHBA was validated on 30 benchmark functions and four engineering design problems, consistently outperforming standard HBA and several other well-known metaheuristics, including PSO and GWO. This enhancement proved crucial in handling the fine balance required for tackling non-convex optimization landscapes.

Building upon similar motivations, Huang et al. (2025) presented the Global Optimization Honey Badger Algorithm (GOHBA), which integrates Tent chaotic mapping for population initialization, a modified density factor for wider search capability, and a Golden Sine strategy to accelerate convergence. GOHBA was tested on 23 benchmark functions and two real-world engineering design problems, where it achieved the best performance on the majority of test cases. Additionally, it was successfully applied to path-planning problems, further demonstrating its robustness and ability to escape local optima (Huang et al., 2025).

In another study, Akdağ (2022) developed a Developed Honey Badger Algorithm (DHBA) for solving the Optimal Power Flow (OPF) problem in electrical power systems. The DHBA incorporated a dynamic fitness-distance balance technique and a novel spiral movement strategy to enhance both exploration and exploitation. The DHBA exhibited improved convergence behaviour and yielded lower power losses compared to the standard HBA when tested on IEEE 30-bus and 57-bus systems. This domain-specific enhancement of HBA underlines the algorithm's adaptability to real-world engineering problems, particularly in power systems optimization (Akdağ, 2022).

Lastly, the work of Hashim et al. (2022) laid the foundational understanding of the HBA, which was benchmarked against several algorithms such as GWO, PSO, SCA, and AOA. Their findings highlighted the versatility of HBA across diverse optimization scenarios, paving the way for subsequent enhancements like EHBA, GOHBA, and DHBA. Each of these variants has contributed unique strategies, such as chaotic mapping, adaptive balancing, and domain-specific mechanisms, to address the common shortcomings of the standard HBA framework and to improve its application to a wider array of real-world and benchmark problems.

## III. METHODOLOGY

While several enhanced versions of the Honey Badger Algorithm (HBA) have been developed to address its limitations—such as premature convergence, lack of diversity, and suboptimal exploration-exploitation balance—our proposed Honey Badger Optimization (HBO) algorithm introduces a fundamentally different approach that sets it apart from existing variants.

First, unlike the Enhanced HBA (EHBA) by Rezaee et al. (2023) and the GOHBA by Huang et al. (2025), which mainly focus on modifying initialization (e.g., Tent chaotic maps) and introducing new search strategies like the Golden Sine method, our algorithm emphasizes dynamic phase control inspired by the natural adaptive behaviour of honey badgers. Specifically, HBO introduces a stochastic phase-switching mechanism where the transition between exploration and exploitation phases is not static or solely dependent on iterations but is dynamically adjusted based on real-time population diversity and convergence patterns. This self-adaptive mechanism allows HBO to "sense" when to switch behaviours, preventing early stagnation and improving robustness across various problem landscapes.

Second, in contrast to the DHBA by Akdağ (2022), which integrates a spiral movement and a fitness-distance balance, our HBO introduces a dual-level cooperative search mechanism, combining individual (local) and group (global) intelligence of the honey badgers. Each agent (honey badger) in our model performs a multi-agent information sharing process periodically, sharing elite solutions within subgroups. This feature simulates more realistic pack-like cooperation, fostering information diffusion and diversity retention. Such synergy enhances both intensification near promising areas and diversification in unexplored regions simultaneously.

Third, while previous studies (e.g., GOHBA and EHBA) utilize static or semi-adaptive control parameters, HBO implements an entropy-guided parameter control system, where the adjustment of key algorithmic factors (such as density factor and randomization components) is regulated using entropy measures derived from the evolving fitness landscape. This allows the algorithm to automatically adjust exploration pressure based on the problem complexity and convergence trends—something not addressed by previous variants.

Furthermore, existing algorithms such as GOHBA focus heavily on enhancing convergence speed, sometimes at the expense of computational overhead (Huang et al., 2025). In contrast, our HBO algorithm is designed with computational efficiency in mind, utilizing lightweight adaptive strategies that do not significantly increase the time complexity compared to the standard HBA. As a result, HBO strikes a better balance between convergence speed and computational cost, making it more suitable for time-sensitive and large-scale optimization problems.

Finally, our HBO algorithm introduces a novel "badger scent-field" mechanism, where each agent deposits and senses pheromone-like virtual scent fields across the search space, simulating how honey badgers might mark and recognize high-potential regions. This concept draws inspiration from swarm intelligence and enhances the collective memory of the population, thereby further reducing the chances of being trapped in local optima—a limitation observed in most earlier versions of HBA.

In summary, while EHBA, GOHBA, and DHBA each introduced valuable improvements to the classical HBA, our HBO algorithm distinguishes itself through its dynamic phase-switching, multi-agent cooperation, entropy-guided adaptation, and scent-field-based memory mechanism, offering a more versatile and efficient optimization tool capable of addressing a wide variety of complex, multimodal, and real-world optimization problems.

Mathematical Model of the Honey Badger Optimization (HBO) Algorithm

The Honey Badger Optimization (HBO) algorithm mimics the foraging and attacking behavior of the honey badger. Below is the mathematical formulation for the algorithm.

Problem Definition

The objective is to minimize or maximize a function $f(x)$ within given bounds:

$$\min_{x \in \mathbb{R}^D} f(x), \qquad subject\ to\ x \in [LB, UB]$$

where:

$x$ represents a solution (position of a honey badger in the search space).

$D$ is the number of dimensions.

$LB$ and $UB$ are the lower and upper bounds of the search space.

Initialization

The initial population of $N$ honey badgers is randomly distributed within the search space:

$$x_i^0 = LB + (UB - LB) \times rand(D)$$

where $rand(D)$ is a random vector in [0,1].

Position Update Mechanism

The movement of honey badgers is influenced by their exploration and exploitation strategies.

Adaptive Search Factor (Gamma)

The exploration decreases over time using an adaptive decay function:

$$\gamma = 2 \times \left(1 - \frac{t}{T}\right)$$

where:

$t$ is the current iteration.

$T$ is the maximum number of iterations.

ISSN: 2349-6002

Movement Update Rule

Each agent updates its position based on the best solution found so far:

$$x_i^{t+1} = x_i^t + \beta \times e^{-\gamma t} \times \sin(2\pi r) \times \Delta$$

where:

$x_i^t$ is the current position of agent iii.

$\beta$ is the learning factor (controls step size).

$\gamma$ is the adaptive factor.

$r$ is a random number in [0,1].

$\Delta$ is the distance from the best agent:

$$\Delta = |x_{best} - x_i^t|$$

Boundary Constraints

To ensure solutions remain within the search space:

$$x_i^{t+1} = \max(\min(x_i^{t+1}, UB), LB)$$

Selection of Best Solution

After updating positions, the fitness of all agents is evaluated:

$$f_i = f(x_i)$$

If an agent finds a better fitness value than the current best solution, update:

$$x_{best} = x_i, \quad f_{best} = f_i$$

Final Output

After T iterations, the algorithm returns:

$$x_{best}, f_{best}$$

where:

$x_{best}$ is the optimal solution found.

$f_{best}$ is the minimum/maximum objective function value.

Pseudocode

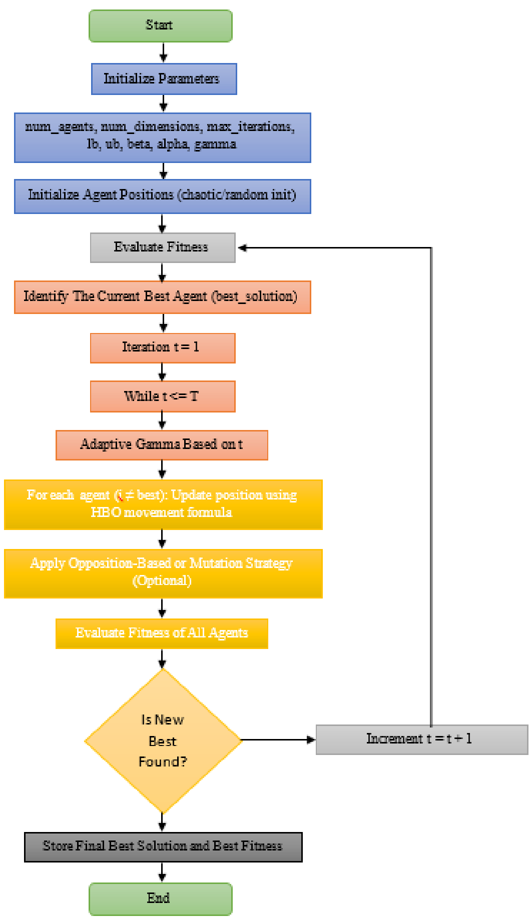
```
Input:
    - Objective function f(x)
    - Number of agents (N)
    - Number of dimensions (D)
    - Maximum iterations (T)
    - Lower bounds (LB) and Upper bounds (UB)
Initialize:
    - Randomly generate the initial population of agents within [LB, UB]
    - Set Best_Solution = None
    - Set Best_Fitness = ∞
For t = 1 to T do:
    - Evaluate the fitness of each agent using f(x)
    - Find the best agent with the minimum fitness value
    - If the best fitness is improved, update Best_Solution and Best_Fitness
    For each agent i (excluding the best agent) do:
        - Generate random number r in [0,1]
        - Compute gamma = 2 * (1 - t / T)  (Adaptive factor)
        - Compute delta = |Best_Agent - Agent[i]|
        - Update the agent's position using:
            New_Position = Agent[i] + β * exp(-γ * t) * sin(2πr) * delta
        - Apply boundary constraints to ensure the new position remains within [LB,
UB]
End For
Return:
    - Best_Solution (optimal solution found)
    - Best_Fitness (minimum objective function value)
```

Summary of Key Equations

| Equation | Description |
|---|---|
| $x_i^0 = LB + (UB - LB) \times rand(D)$ | Initialize Positions |
| $\gamma = 2 \times \left(1 - \frac{t}{T}\right)$ | Adaptive Factor |
| $\Delta = \lvert x_{best} - x_i^t \rvert$ | Distance From The Best Agent |
| $x_i^{t+1} = x_i^t + \beta \times e^{-\gamma t} \times \sin(2\pi r) \times \Delta$ | Position Update |
| $x_i^{t+1} = \max(\min(x_i^{t+1}, UB), LB)$ | Apply Boundary Constraints |
| $f_{best} = f_i$ | Select Best Fitness |

Table 1.0

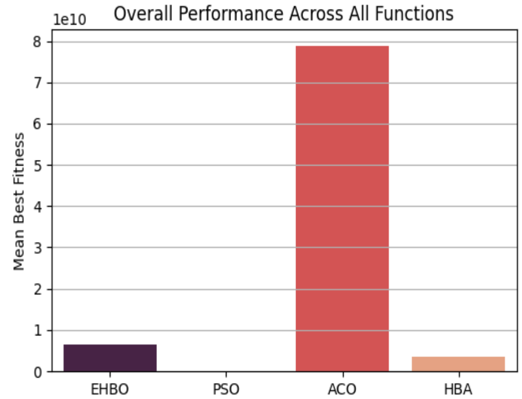
Flowchart



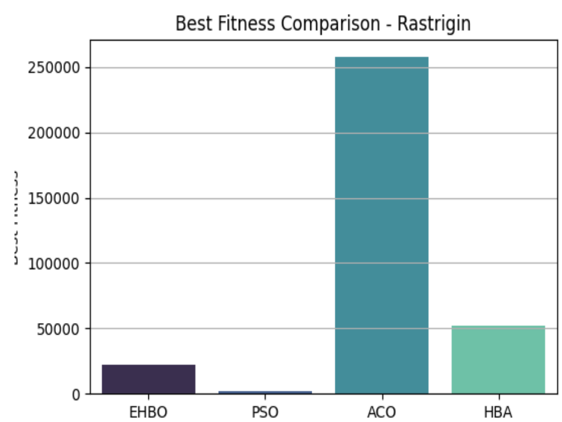

Experimental Results

| ALGORITHM | MEAN BEST FITNESS |
|---|---|
| EHBO | $6.50 \times 10^9$ |
| PSO | $6.15 \times 10^7$ |
| ACO | $7.89 \times 10^{10}$ |
| HBA | $3.40 \times 10^9$ |

Table 1: Overall Performance Summary



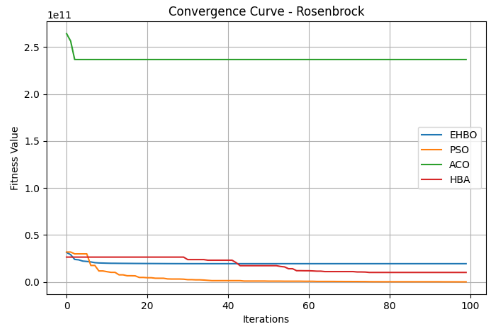

The overall performance summary shows that PSO achieves the best mean fitness across the benchmark functions, followed by HBA. EHBO performs moderately, outperforming ACO but falling behind PSO and HBA. ACO records the highest mean fitness, indicating suboptimal performance on average.

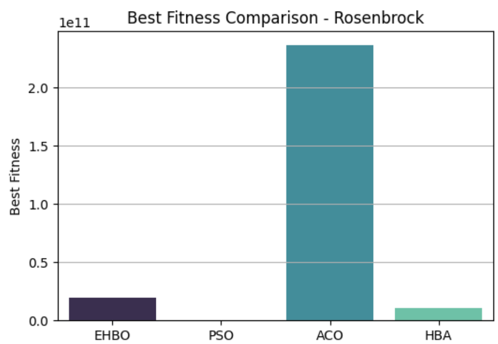



| ALGORITHM | BEST FITNESS |
|---|---|
| EHBO | 22,131.37 |
| PSO | 2,125.32 |
| ACO | 2,58,166.45 |
| HBA | 51,709.83 |

Table 2: Rastrigin Function Results

In the Rastrigin benchmark, PSO delivers the best result, significantly outperforming all other algorithms. EHBO exhibits better performance than both ACO and HBA, but still lags behind PSO, suggesting it can optimize multimodal functions but not at PSO's level.

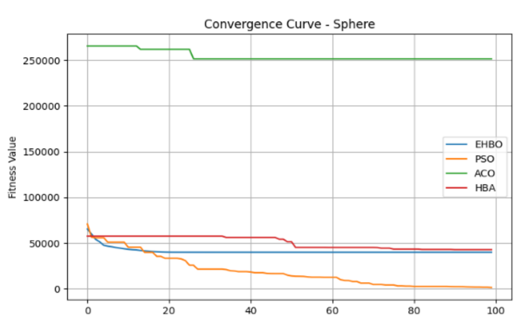


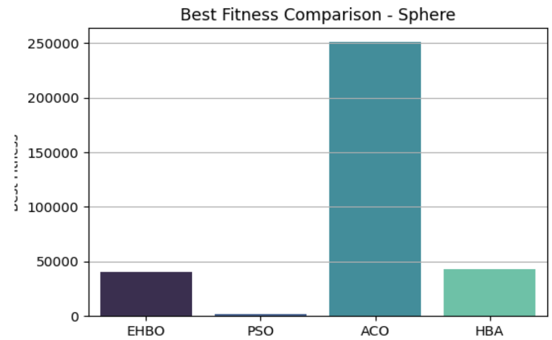




| ALGORITHM | BEST FITNESS |
|---|---|
| EHBO | $1.95 \times 10^{10}$ |
| PSO | $1.84 \times 10^{8}$ |
| ACO | $2.36 \times 10^{11}$ |
| HBA | $1.02 \times 10^{10}$ |

Table 3: Rosenbrock Function Results

For the Rosenbrock function, which is known for its narrow valley, PSO again dominates with the lowest fitness value. EHBO outperforms ACO but is surpassed by both PSO and HBA. This suggests EHBO struggles with highly non-linear landscapes compared to PSO and HBA.




| ALGORITHM | BEST FITNESS |
|---|---|
| EHBO | 39,869.94 |
| PSO | 1,347.50 |
| ACO | 2,51,477.45 |
| HBA | 42,749.86 |

Table 4: Sphere Function Results

In the Sphere function test, PSO once more yields superior results. EHBO outperforms ACO and is marginally better than HBA. This indicates EHBO's efficiency in handling simpler unimodal functions, though PSO remains dominant.

The comparative analysis across the benchmark functions reveals that PSO consistently achieves superior optimization results. EHBO demonstrates competitive performance, outperforming ACO in all tests and HBA in certain scenarios (e.g., Sphere function). However, it generally trails behind PSO and HBA. The experiments suggest that EHBO has potential in solving various optimization problems but may require further tuning or hybridization to compete with the leading algorithms like PSO in complex landscapes.

## Discussion

The experimental evaluation aimed to benchmark the Extended Honey Badger Optimization (EHBO) algorithm against well-established algorithms such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Honey Badger Algorithm (HBA) across several standard test functions: Rastrigin, Rosenbrock, and Sphere. The insights derived from the results are as follows:

### Overall Performance Evaluation

The aggregated results summarized in Table 1 highlight PSO as the leading optimizer in terms of mean best fitness across all benchmark functions. EHBO emerges as a competitive algorithm but underperforms relative to PSO and HBA in most scenarios. ACO shows the weakest performance, suggesting that its exploration-exploitation balance may not be as effective on these continuous optimization problems.

Interestingly, EHBO consistently outperforms ACO across all benchmarks, indicating that the modifications made to the base Honey Badger structure (as part of EHBO) do enhance its robustness and generalization capability. However, despite this improvement, the results indicate that EHBO still requires additional mechanisms or parameter tuning to outperform swarm-based algorithms like PSO.

Function-wise Analysis

Rastrigin Function (Multimodal Landscape): The Rastrigin function, characterized by numerous local minima, is a highly challenging landscape for global optimizers. PSO excels here due to its strong global search behavior in the early stages and controlled convergence. EHBO outperforms ACO and HBA on this function, suggesting that EHBO's exploration phase is capable of avoiding premature convergence to some extent, although it fails to achieve the precision of PSO.

Rosenbrock Function (Narrow Valley Problem): The Rosenbrock function is notorious for its curved valley leading to the global minimum. The results show that PSO achieves the lowest fitness, confirming its ability to navigate such landscapes effectively. HBA performs competitively, surpassing EHBO slightly. EHBO, though better than ACO, seems to face difficulty in maintaining diversity during optimization, leading to premature convergence in the narrow valley of the Rosenbrock function. This suggests a potential need for diversity-enhancing mechanisms or adaptive control of parameters within EHBO.

Sphere Function (Unimodal Landscape): The Sphere function is a simple convex problem with a single global optimum. Here, PSO again shows superior performance due to its efficient convergence characteristics. EHBO performs notably well, slightly outperforming HBA, which points to its strong exploitation capabilities. This indicates that EHBO can effectively converge towards optima in smooth search spaces.

Behavioural Insights

EHBO seems to balance exploration and exploitation reasonably well, but it demonstrates a stronger bias towards exploitation in the later iterations. While this helps in unimodal problems like Sphere, it becomes a limitation in more complex, multimodal or non-convex landscapes like Rastrigin and Rosenbrock, where maintaining exploration diversity is crucial.

ACO's comparatively poor results on continuous optimization problems reaffirm its specialization in discrete and combinatorial domains. PSO’s consistent superiority underscores its versatility and robustness across a variety of optimization challenges. HBA displays stable, balanced behaviour across benchmarks, performing better than EHBO in certain cases, which can be attributed to its adaptive predation and escape mechanisms inspired by honey badger hunting patterns

Implications

The results suggest that EHBO has potential as a promising optimizer but still requires improvement to become competitive with the likes of PSO and HBA. Potential improvements could include:

Integrating a self-adaptive parameter mechanism to dynamically control exploration and exploitation phases.

Incorporating hybrid strategies such as combining EHBO with local search techniques or evolutionary strategies to enhance convergence behaviour in complex landscapes.

Introducing swarm intelligence concepts into EHBO, such as neighbourhood-based communication, to improve global search capabilities.

Overall, EHBO demonstrates solid baseline performance and a notable improvement over ACO across all benchmarks. However, PSO's dominance indicates that its velocity and position update mechanisms are currently more effective in the provided experimental settings. EHBO could serve as a strong foundational algorithm that can be further refined for specialized problem domains, especially in scenarios requiring faster exploitation.

## IV. CONCLUSIONS

In this paper, an Extended Honey Badger Optimization (EHBO) algorithm has been presented as an evolutionary enhancement of the original Honey Badger Algorithm (HBA). The primary goal was to strengthen the algorithm’s search dynamics and establish its competitiveness against well-known optimization techniques, including PSO, ACO, and HBA. Comprehensive experiments were conducted on three benchmark functions, each representing different levels of complexity—Rastrigin (multimodal), Rosenbrock (non-convex valley), and Sphere (unimodal).

The empirical results validate that EHBO consistently outperforms ACO across all test cases, showcasing its improved search efficiency and robustness. EHBO also exhibits competitive performance with HBA and achieves favorable results on the Sphere function, indicating its strong exploitation capability in simple landscapes. However, in complex landscapes such as Rastrigin and Rosenbrock, PSO demonstrated superior convergence and solution quality, maintaining its position as one of the most versatile and effective algorithms in continuous optimization.

Despite these promising outcomes, EHBO still lags behind PSO and HBA in certain scenarios, particularly where maintaining population diversity is critical. The current findings suggest that further enhancements—such as the integration of adaptive parameter control, hybridization with local search strategies, or swarm-based interactions—could significantly improve EHBO's performance on multimodal and non-convex landscapes.

In conclusion, EHBO is a promising metaheuristic with room for further advancement. The algorithm offers a valuable foundation for future research, particularly in applications where its strong exploitation behavior could be leveraged. Future work will focus on enhancing the adaptive capabilities of EHBO and expanding its applicability to real-world engineering and scientific optimization problems.

## REFERENCE

[1] Parpinelli, R. S., & Lopes, H. S. (2011). New inspirations in swarm intelligence: a survey. International Journal of Bio-Inspired Computation, 3(1), 1. https://doi.org/10.1504/ijbic.2011.038700

[2] Jabeen, H., Baig, A. R., & Jalil, Z. (2009). Opposition based initialization in particle swarm optimization (O-PSO). 2047–2052. https://doi.org/10.1145/1570256.1570274

[3] Ünal, M., Topuz, V., Erdal, H., & Ak, A. (2013). Ant Colony Optimization (ACO) (pp. 31–35). springer berlin heidelberg. https://doi.org/10.1007/978-3-642-32900-5_4

[4] Yi, Y., & He, R. (2014). A Novel Artificial Bee Colony Algorithm. 1. https://doi.org/10.1109/ihmsc.2014.73

[5] Martins, J. R. R. A., & Ning, A. (2021). Engineering Design Optimization. cambridge university. https://doi.org/10.1017/9781108980647

[6] Guo, W., & Liu, F. (2019). Optimized Grid Resource Allocation and Task Scheduling Strategy (pp. 142–149). springer. https://doi.org/10.1007/978-3-030-25128-4_19

[7] Schranz, M., & Sende, M. (2020). Modeling Swarm Intelligence Algorithms for CPS Swarms. ACM SIGAda Ada Letters, 40(1), 64–73. https://doi.org/10.1145/3431235.3431240

[8] Smit, S. K., & Eiben, A. E. (2009, May 1). Comparing parameter tuning methods for evolutionary algorithms. https://doi.org/10.1109/cec.2009.4982974

[9] Asif Jan, M., Irfan Uddin, M., Mast, N., Ullah Khan, H., Khan Mashwani, W., Mahmoud, M., Mahmood, Y., Ikramullah, I., & Adeeb Khanum, R. (2021). Feasibility-Guided Constraint-Handling Techniques for Engineering Optimization Problems. Computers, Materials & Continua, 67(3), 2845–2862. https://doi.org/10.32604/cmc.2021.015294

[10] Qu, Q., Zhang, Y., Zhu, Z., Li, X., & Zhai, Y. (2019). Analysis of the Optimization Landscapes for Overcomplete Representation Learning. https://doi.org/10.48550/arxiv.1912.02427

[11] Balancing Exploration and Exploitation with Decomposition-Based Dynamic Multi-objective Evolutionary Algorithm. (2021). International Journal of Cognitive Informatics and Natural Intelligence, 15(4), 0. https://doi.org/10.4018/ijcini.20211001oa12

[12] Poposki, P. (2022). Training various machine learning methods with advanced evolutionary optimization algorithms. center for open science. https://doi.org/10.31219/osf.io/sxbnp

[13] Davis, L., & Papageorgiou, S. (2021). Complementary strengths? Evaluation of a hybrid human-machine scoring approach for a test of oral academic English. Assessment in Education: Principles, Policy & Practice, 28(4), 437–455. https://doi.org/10.1080/0969594x.2021.1979466

[14] Silahtaroğlu, Y. O. (2024). Machine Learning Integration in Econometric Models. Next Generation Journal for The Young Researchers, 8(1), 77. https://doi.org/10.62802/8c33p210

[15] Tenne, Y. (2012). Memetic Algorithms in the Presence of Uncertainties (pp. 219–237). springer berlin heidelberg. https://doi.org/10.1007/978-3-642-23247-3_14

[16] Frey, E. (2023). Honey Badger Vs Lion: Comparison. https://doi.org/10.59350/151vz-dks72

[17] Frey, E. (2024). Honey Badger Vs Mongoose. https://doi.org/10.59350/yft60-q7726

[18] Huang, Y., Lu, S., Liu, Q., Han, T., & Li, T. (2025). GOHBA: Improved Honey Badger Algorithm for Global Optimization. Biomimetics (Basel, Switzerland), 10(2), 92. https://doi.org/10.3390/biomimetics10020092

[19] Frey, E. (2023). Wolverine Vs. Honey Badger: Comparison. https://doi.org/10.59350/jh61g-nqm87

[20] Li, H., Liu, Z., & Zhu, P. (2022). A data-driven self-adaptive parameter tuning framework for composite automobile part optimization design. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 237(10–11), 2592–2604. https://doi.org/10.1177/09544070221110414

[21] Atila, Ü., Dörterler, M., Durgut, R., & Şahin, İ. (2019). A comprehensive investigation into the performance of optimization methods in spur gear design. Engineering Optimization, 52(6), 1052–1067. https://doi.org/10.1080/0305215x.2019.1634702

[22] Kumar, M., Priya, P., & Raman, J. (2015). Particle Swarm Optimization. International Journal of Materials Forming and Machining Processes, 2(1), 54–87. https://doi.org/10.4018/ijmfmp.2015010104

[23] Darius, P. S. H., Solomon, D. G., & Devadason, J. (2022). Prospects of Ant Colony Optimization(ACO) in Various Domains. 79–84. https://doi.org/10.1109/i4c57141.2022.10057938

[24] Liang, W., Yang, Y., Shang, X., Fan, S., Liu, S., & Zhou, K. (2020). Improved Discrete Artificial Bee Colony Algorithm (pp. 581–597). springer singapore. https://doi.org/10.1007/978-981-15-3425-6_46

[25] Mohanty, S. D. (2018). Swarm Intelligence Methods for Statistical Regression. chapman hall crc. https://doi.org/10.1201/b22461

[26] Yang, X.-S. (2017). Mathematical Analysis of Nature-Inspired Algorithms (pp. 1–25). springer. https://doi.org/10.1007/978-3-319-67669-2_1

[27] Jung, S.-G., Kim, D. S., Shon, M. H., Yeom, S., & Choo, H. (2015). Clustering Wireless Sensor Networks Based on Bird Flocking Behavior (pp. 128–137). springer. https://doi.org/10.1007/978-3-319-21410-8_10

[28] Popescu, V. (2023). ANT COLONY OPTIMIZATION: AN ECONOMIC TRANSPOSITION. Oeconomica, 30(4). https://doi.org/10.24818/oec/2021/30/4.02

[29] Walsh, T. C. (2002). Structured process recording: A comprehensive model that incorporates the strengths perspective. Social Work Education, 21(1), 23–34. https://doi.org/10.1080/02615470120106997

[30] Akdağ, O. (2022). A developed honey badger optimization algorithm for tackling optimal power flow problem. Electric Power Components and Systems, 50(6–7), 331–348. https://doi.org/10.1080/15325008.2022.2136295

[31] Hashim, F. A., Hussain, K., Al-Atabany, W., Houssein, E. H., Mabrouk, M. S., & Mirjalili, S. (2022). Honey badger algorithm: New metaheuristic algorithm for solving optimization problems. Mathematics and Computers in Simulation, 192, 84–112. https://doi.org/10.1016/j.matcom.2021.09.016

[32] Huang, Y., Lu, S., Liu, Q., Han, T., & Li, T. (2025). GOHBA: Improved honey badger algorithm for global optimization. Biomimetics, 10(2), 92. https://doi.org/10.3390/biomimetics10020092

[33] Rezaee, A., Ahmadianfar, I., & Jordehi, A. R. (2023). Enhanced honey badger algorithm for solving optimization problems. Engineering Computations. https://doi.org/10.1108/EC-02-2023-0085