**An Optimal Performance of Normal and Modified Honey Badger Algorithms on Unimodal Test Functions**

A. Yadava), J. Kaur,b)  
*Chandigarh University, Gharuan, Mohali, India.*a)Corresponding author: *Anjali.e6925@cumail.in*  
b)*Jatinder.e1486@cumail.in*

Abstract: Metaheuristic optimization algorithms are widely employed to solve complex engineering and scientific problems, and their effectiveness can often be assessed using benchmark test functions. This study evaluates the performance of the original Honey Badger Algorithm (HBA) and several of its modified variants on unimodal test functions, which are designed to test the exploitation ability of optimization methods. Unimodal functions, characterized by a single global optimum, provide a focused environment to assess an algorithm's local search accuracy and convergence behavior. The comparative analysis involves performance metrics such as convergence speed, solution accuracy, and robustness across multiple runs. Results indicate that while the standard HBA exhibits strong exploitation capabilities, the modified versions—incorporating enhancements such as chaotic maps, adaptive parameters, and hybrid strategies—consistently outperform the original in terms of convergence precision and stability. The study confirms the potential of improved HBA variants for high-precision optimization tasks involving unimodal landscapes.

# **Introduction**

Optimization, as described by *Jain et al. (2019)*, is the methodical pursuit of the best solution to a given problem using various methodologies. This technique is critical for increasing competitiveness, lowering costs, and enhancing system performance in all engineering application including business, engineering, and manufacturing, and medicine. As of societal complexity rises, rises the challenges and significance of optimization.

Traditional optimization algorithms, which frequently use predictable and gradient-based methods, perform well for single-peaked functions. Nevertheless many real-world engineering challenges are nonlinear, making standard techniques ineffective. Metaheuristic algorithms, which are developing as innovative computational tools, provide alternate approaches to addressing these challenging problems.

*Kumar and Vohra (2021)* define metaheuristic algorithms as optimization strategies based on metaheuristic information. They use heuristic approaches to analyses and find solutions, extracting crucial characteristics using meta-feature inspection. This allows for the development of more effective optimization solutions, especially in complicated, high-dimensional environments where standard methods fail.

Metaheuristics are preferred because of their capacity to maximize issue attributes, minimize solution pressure and limit the search size, and base repetitions on empirical evidence. They efficiently investigate the field and objective function, take into account restrictions and external factors, and adapt to changing conditions, hence enhancing problem-solving abilities. Essentially, they provide a solid and extensible framework for identifying optimal solutions.

Metaheuristic algorithms priorities problem-solving speed and practicality. They thrive in quickly coming up with good, if not ideal, solutions. Their focus on efficiency and application has resulted in their extensive use in optimization.

Researchers frequently employ algorithms like the Ant Lion Optimizer (ALO), Particle swarm optimization (PSO), crow search algorithm (CSA), Wild Horse Optimizer (WHO), and Artificial Bee Colony (ABC). These algorithms, due to their inherent characteristics, that are suitable for tackling composite optimization difficulties where speed then practical solutions are paramount. Their ability to navigate large search spaces efficiently makes them valuable tools in various domains.

The honey badger algorithm (HBA), influenced by honey badger foraging, has grown in popularity due to its simple design and good performance. Its potential for a variety of applications is enormous. Researchers are continually attempting to increase the HBA's optimization capabilities.

Efforts to improve HBA include incorporating differential evolution theory (Dong et al.), merging it with Genetic Algorithm (GA) operators and Levy flight strategies (Deng et al.), and introducing multi-strategy upgrades (Xiang et al. These multi-strategy techniques include adaptive weighing, starved search tactics, and limited inverse learning processes. These changes attempt to improve HBA's algorithm for searching and overall optimization efficiency.

Researchers have explored various modifications to enhance the Honey Badger Algorithm (HBA). Han et al. (2022) proposed integrating reverse learning and chaos mechanisms to improve population quality and procedure efficiency. Similarly, Yasear et al. utilized a reverse learning machinery to enhance HBA, smearing it to solve complex power consumption cost problems.

*Nassef et al. (2022)* adopted a shooting examination method, which increased HBA's optimisation potential and allowed for an easier adjustment among exploration and exploitation. Lei et al. (2022) made various improvements, including using the law of quasi-cosines for updating the density aspect, implementing a pinhole imaging approach to increase population variety, and creating a spiral search mechanism for enhanced global search capability during the stage of exploration.

*Düzenli et al. (2022)* tackled the PV estimate of parameters problem using adversarial learning and Gaussian chaotic mapping to handle critical unpredictable variables within HBA. These changes try to solve the original HBA's weaknesses through enhancements to population diversity, search productivity, and convergence speed.

*Dao et al. (2023)* enhanced the Honey Badger Algorithm (HBA) by combining multi-directional approaches with elite reverse learning, and effectively applied it to the Wireless Sensor Network (WSN) node coverage problems.

Furthermore, spiral curves, which can be found both spontaneously and artificially, are increasingly being employed to create optimization algorithms. These spirals provide distinctive search patterns, allowing algorithms to effectively navigate complex search regions. Their adoption can have a major impact on an algorithm's capability to find optimal solutions, emphasizing the essential role of functional and geometric principles in enhancing optimization approaches. This demonstrates the interaction of fundamental mathematical concepts with advanced optimization methodologies.

Spiral patterns, which appear in nature and mathematical constructions, are increasingly being used to improve metaheuristic optimization methods. The Moth-Flame Optimization (MFO) technique, for example, represents a moth's flight route as a spiral calculation of a blaze (Mirjalili 2015). Similarly, the Whale Optimization Algorithm (WOA) uses spiral motion to replicate humpback whale hunting behavior.

*Sun et al. (Hao et al. 2022)* studied the result of examination pathways on WOA presentation, proving the efficacy of spiral-based upgrades through experimental comparison. Guo et al. (2020b) presented an Ant Lion Optimizer (ALO) that uses a spiral complicated route search model and eight different spiral path search algorithms to progress variation in populations and equilibrium extraction and exploration.

*Zhang et al. (2023)* designed an MRFO algorithm that employs an algebraic spiral foraging strategy. This modification increases the algorithm's global search abilities, speeds up convergence, besides displays success in power system economics. These research show the importance of spiral-based searches in increasing metaheuristic algorithms' performance and efficiency, as well as their capacity for navigating complex search environments and identify optimal solutions.

**Optimization Models**

**Honey Badger Optimisation (HBO)**

Honey Badger Optimization (HBO) is a recently developed nature-inspired optimization algorithm based on honey badgers' foraging and digging behaviors.

It belongs to the group of metaheuristic algorithms designed to tackle complicated optimization issues by replicating the adaptable and intelligent behavior of animals in nature.

**Key Concepts of HBO:**

**Foraging Behavior:** Honey badgers' keen sense of smell helps them find hidden prey such as honey and tiny animals. This is analogous to the exploration step in optimisation, in which the algorithm explores widely through the solution space.

**Digging Behavior:** Honey badgers employ persistent digging to reach food, similar to how optimisation algorithms refine viable solutions around attractive locations.

**Dynamic Transition:** The HBO method balances exploration (wide search) with exploitation (intense local search) to achieve effective convergence to the global optimum and prevent local minima.

**Algorithm Process:**

**Initialization:** Generate a random set of candidate solutions from the search space.

**Evaluate** each candidate's fitness using the objective function.

**Exploration (Global Search):** Model the honey badger's activity by randomly perturbing algorithms to discover new places.

**Exploitation (Local Inquiry):** Focus on the greatest solutions and intensify surrounding them to fine-tune the search.

**Updating Solutions:** Use probabilistic principles to replicate honey badger behaviour and adapt the search method accordingly.

**Stopping Criteria:** End the process when the extreme numeral of iterations is exceeded or the clarification meets an acceptable error level.

**Pseudo code (Honey Badger Optimization (HBO):**

BEGIN HoneyBadgerOptimization

Initialize population of honey badgers (positions) randomly

Define problem constraints and objective function f(x)

Set algorithm parameters (max\_iterations, population\_size, etc.)

Calculate fitness for each badger (evaluate f(x))

WHILE (termination condition not met)

FOR each honey badger i

Select a random badger j from the population

IF (rand() < digging\_probability)

Exploitation phase - Digging

Update position using digging formula:

ELSE

Exploration phase - Searching

Update position by moving randomly:

END IF

Apply boundary constraints on

Evaluate new fitness f()

IF f() < f()

Update =

END IF

END FOR

Update control parameters α, β, γ dynamically

Store best solution found so far

END WHILE

RETURN X\_best (optimal solution)

END HoneyBadgerOptimization

Wireless Sensor Networks (WSNs) are comprised of a large number of battery-powered units that are interconnected to facilitate data processing, control, collection, and communication. Originating from military applications, WSNs have undergone significant research and development, resulting in their widespread adoption across diverse sectors, including agriculture, disaster relief, and infrastructure monitoring. The benefits of WSNs over traditional wired networks include reduced cabling requirements, enhanced mobility, lower installation costs, and increased automation capabilities.

Within a WSN, each node is a multi-functional unit capable of sensing, data gathering, processing, and communication with neighboring nodes or a sink. The physical resources available to each node directly impact the frequency, size, and quality of the sensed data. Therefore, flexibility, cost-effectiveness, and energy efficiency are crucial design considerations

**Sensor Node**

A sensor node's functionality relies on a combination of sensor units and essential components, including a microcontroller, signal conditioning, transceiver, and power management modules. These nodes, designed for energy efficiency and multi-functionality capture environmental data using sensors that measure parameters such as temperature and pressure. The gathered data is transmitted from individual sensor nodes to a central sink node, and finally, via a gateway, to the intended destination.

The Node, an essential element of a WSN, includes a memory for data storage, a battery for power, and an Analog-to-Digital Converter (ADC) to connect with sensors and other devices, enabling ad-hoc networking. For a WSN to function correctly, motes must collect and share information. A sensor node is created when a mote is combined with a sensor

**Network size**

The scope of the sensor field and the network's topology directly impact the size of a WSN. The distance spanning the network, determined by the transmission range of the sensor nodes, serves as a measure of its size. This distance is essentially the shortest path connecting the two most remote nodes.

**Node density**

Node density indicates the presence of a sufficient number of sensor nodes within a given area, such that each node remains within the communication range of several other nodes at all times. This attribute is critical for maintaining robust connectivity within a Wireless Sensor Network (WSN). It is a fundamental requirement in sensor network exploration to ensure that node density is adequately high, thereby guaranteeing that all nodes are continuously within the communication range of multiple neighboring nodes. As demonstrated by Zhang. X [1], this density can reach levels as high as 20 sensor nodes per square meter.

**Node capabilties**

The operational capabilities of sensor nodes are constrained by limitations in control, computational resources, memory, and power. Despite the integration of sensing and registration devices, radio transceivers, and energy components, these nodes exhibit restricted processing speed, storage capacity, and communication bandwidth

**Communication Mode**

The communication capabilities of sensor nodes are fundamental to the operation of a network model. The high density of sensor nodes results in close proximity between neighboring nodes, which in turn influences the overall performance of the system. To optimize energy usage, multi-hop communication is typically favored over single-hop communication in WSNs. Ultimately, the quality of a WSN is determined by the robustness of its communication model.

**Topology**

The architecture of a sensor network, known as its topology, is contingent upon a multitude of design factors. Mobile sensor nodes are particularly influential in inducing topological alterations. The network's topology is a key determinant of system performance. In Wireless Sensor Networks (WSNs), topologies are employed to establish dynamic networks that facilitate data collection and processing, with mesh, star, and tree configurations being commonly utilized.

**Self organization**

The autonomous nature of sensor nodes in WSNs, as demonstrated by their node localization algorithms and protocols, introduces a set of design constraints. These constraints necessitate careful consideration during the development of protocols and algorithms for these networks.

The ability to accurately localize sensor nodes is a fundamental requirement for many applications within Wireless Sensor Networks (WSNs), as it enables the optimization of various parameters, leading to increased node lifetime and improved accuracy. The existing body of literature offers comprehensive explanations of diverse localization methodologies, which generally adhere to a two-stage process: the initial stage involves the measurement of distances, followed by a computational stage to determine node positions.

**Anchor-free and anchor-based Localixation**

The localization of target nodes is accomplished through techniques that either utilize known anchor node positions or rely on relative coordinate information. Anchor-based methods, which require GPS-equipped anchors, calculate distances and apply algorithms to determine unknown node coordinates. A minimum of three anchors is necessary for 2D and four for 3D localization. Anchor-free methods circumvent the need for anchor positions by using relative coordinate data.

**Distributed and Centralized Localization**

Centralized localization algorithms rely on a central processor for all computations, contrasting with distributed methods that use local sensor data. While this approach requires all nodes to send data to the base station, it allows algorithms like simulated annealing and RSSI-based localization to achieve higher accuracy due to the availability of complete network connectivity and distance information.

**Static and Dynamic Localization**

Static localization algorithms are designed for stationary sensor nodes, with coordinates determined once during deployment. Dynamic algorithms handle mobile nodes, necessitating continuous tracking and coordinate updates. Dynamic localization is more complex and time-consuming, but static algorithms offer faster convergence. Range-based techniques like triangulation and centroid methods are examples of static algorithms.

**Determining 2D and 3D coodinates of a sensor node**

The significant strides made in wireless communication technologies in recent years have catalyzed the widespread adoption of Wireless Sensor Networks (WSNs) across a multitude of real-world applications. In this context, the localization of sensor nodes has emerged as a critical requirement for virtually all WSN deployments. The existing body of literature offers a diverse array of localization algorithms, providing solutions for determining the location of unknown nodes, with methodologies that accommodate both the presence and absence of anchor nodes.

**2D coordinates of a sensor node:***Zhang. X [1]* conducted research on outdoor localization in the absence of GPS, employing the centroid method to compute the coordinates of unknown nodes. Saeed N et al. [2] focused on minimizing the number of anchor nodes required for localization, achieving high accuracy in distance estimation by prioritizing the determination of shorter distances. Chelouah L et al. [3] proposed alternative methods for distance estimation, utilizing both statistical techniques and neural network concepts, and identified key parameters, such as transmitted power, radio frequencies, and node mobility, that can negatively impact the accuracy of distance measurements.

*Parulpreet et al. [4]* explored multi-alteration, proposing a distributed approach where sensor nodes collaboratively solve optimization problems beyond individual capabilities. Their method involved three stages: a combined initial step, initial estimate calculation, and subsequent location refinement. They also developed a distributed localization technique for ad-hoc networks, emphasizing its stability and balanced power consumption compared to centralized methods, which, though more accurate, can be less efficient. Singh, P, Khosla et.al. [5] Focusing on static target nodes, employed the least squares approach using RSSI measurements and a single anchor node. This method aimed to achieve accurate position determination. These studies highlight the diverse strategies employed in WSN localization, balancing accuracy, efficiency, and network stability.

*Sharma G et al. [6]*explored range-free strategies that are energy efficient and use fewer anchor nodes in their conclusions. They believed that by using less anchor nodes, their technique is less complicated and more effective. Salgotra, R. et al. [7] suggested a method for real-time localization that is both efficient and stable. The algorithms used are known as the Breadth-First (BRF) and Backtracking Greedy (BTG). Nguyen T et al. [8] proposed a procedure that falls within the category of range-free methods. Their work claimed that their scheme is simple and realistic, in which a comparison among the nodes is based on RSSI values, which consumes far less energy and shows great accuracy by using mobile anchor concepts. Brest et al. [9] suggested two algorithms for providing mobility in WSNs. Rohit Salgotra et al [10] addressed anchor-based localization algorithms and computed the efficiency and accuracy of localization in relation to anchor mobility and target nodes. There is much more study that we have not described, and the results can be found in the literature.

These studies, as a whole, demonstrate a wide range of strategies employed in WSN localization, encompassing energy-efficient range-free techniques, real-time algorithms, and solutions tailored for mobile environments. Researchers are constantly working to find the optimal balance between accuracy, efficiency, and power usage, tackling the specific problems that arise in different WSN setups.

**3D coordinates of a sensor node based on range based approaches:** *Salgotra R et al. [11]* leveraged Ultra-Wide Band and Time-of-Arrival (ToA) to achieve precise 3D coordinate localization, demonstrating its superior accuracy over existing methods. W. Xia et al. [12] introduced a complex, high-cost DvHop scheme for efficient 3D sensor node localization. Sana Shahab et al. [13] proposed a hybrid optimization approach, combining Distance Vector hopping with the Newton method, focusing on enhancing coverage and accuracy.

*V. Ngunzi et al. [14]* developed an RSSI-based model for 3D coordinate determination in WSNs, establishing a correlation between the Degree of Irregularity and transmitted signal range variance. J.A. Ternero et al. [15] employed a parametric algorithm for 3D localization, optimizing performance with minimal anchor nodes by converging the network towards a central point.

These studies highlight diverse strategies for 3D localization, ranging from UWB-based precision to hybrid optimization techniques. Each approach addresses specific challenges, balancing accuracy, complexity, and cost. The research collectively advances 3D localization capabilities, offering solutions tailored to various WSN deployment scenarios. The studies demonstrate a continuous effort to refine techniques, improve accuracy, and reduce deployment costs in 3D WSN localization.

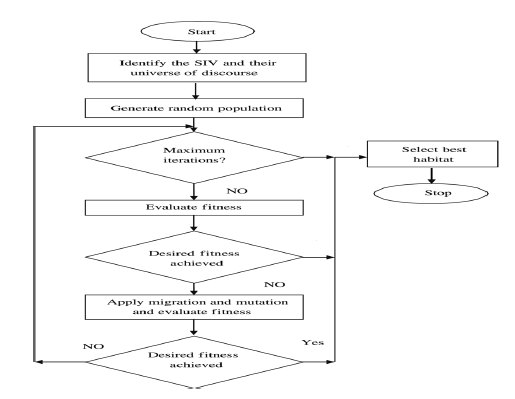
*L.K. Ketshabetswe et al. [16]* proposed a beacon-signal-dependent localization algorithm, albeit with higher cost. Existing 2D methods like MDS-MAP, DVHOP, and Centroid have been adapted for 3D localization, as documented in various studies. J. Prakash et al. [17] and Huang Z Y et al. [18] surveyed 3D localization in underwater networks, where connectivity and anchor node count are crucial for position determination.

*Y. Qiu et al. [19]* developed a hybrid approach for 3D WSNs, utilizing an approximation based on the least squares criterion. These studies demonstrate the ongoing evolution of 3D localization techniques, addressing diverse application needs. D.M. John et al. [20] and L.K. Ketshabetswe et al. [21] suggested adapting and combining existing methods to enhance accuracy and efficiency, particularly in challenging environments like underwater networks.

**Techniques to improve localization**

**Features of optimization techniques:** The optimization techniques employed in localization systems must possess several critical features to ensure effective performance. Firstly, speed is paramount, as real-time environments demand rapid convergence of the localization algorithm. Secondly, adaptability is essential, enabling the system to respond to environmental changes, such as node failures, and adjust accordingly. Finally, self-organizing capability is crucial, allowing the system to autonomously reconfigure the network in response to mobility or other dynamic changes, guided by pre-defined rules and instructions.

**Criteria for evaluation and paramter calculation:** Wireless Sensor Networks (WSNs) are prone to a variety of errors that can significantly impact their performance, including errors stemming from range measurement inaccuracies, the unavailability of GPS signals, and instances where localization algorithms themselves contribute to a reduction in system accuracy. Range errors originate from incorrect distance measurements, while GPS errors are attributed to inaccuracies in anchor node positions. The performance of localization algorithms is evaluated based on several key parameters, including location accuracy, cost-effectiveness, and coverage area, as outlined in Figure 1



**FigURE 1,** Flowchart of Firefly algorithm

**Accuarcy in location:**The accuracy of a localization algorithm is evaluated by calculating the discrepancy between a node's true position and its estimated location; a smaller deviation signifies higher accuracy. Additionally, the algorithm must exhibit flexibility to mitigate errors and noise introduced from the input. Coverage is directly proportional to the number of anchor nodes deployed within the sensor field, with a higher count resulting in improved coverage. Finally, cost is assessed based on the power consumed and the time required by the algorithm to localize nodes, which are critical factors for initiating communication between them.

**Parameter calcualtion on the basis of accuracy:** To evaluate localization accuracy, the estimated target node positions are compared to their actual positions. The resulting errors, representing the difference between these positions, are typically measured using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

**The Mean Absolute Error (MAE)** serves as a quantitative measure of the accuracy of localization algorithms, particularly in applications involving continuous variables. It computes the average of the absolute differences between the actual and calculated positions of sensor nodes.

**Root Mean Square Error (RMSE**) quantifies localization accuracy, measuring the average magnitude of errors. It calculates the square root of the mean of squared differences between actual and estimated node positions. A lower RMSE indicates higher accuracy, reflecting smaller deviations between predicted and true locations. Essentially, it provides a robust measure of the algorithm's precision*.*

**Localizationof mobile nodes in wireless sensor networks using DA-FA algorthim**

WSNs enable global environmental data collection, but encounter performance hurdles like localization, network longevity, task scheduling, routing, security, and deployment. To overcome these, Computational Intelligence (CI) techniques have become increasingly prevalent. CI algorithms offer robust and autonomous 2D or 3D node localization, minimizing hardware dependencies. These algorithms excel at accurately determining the positions of static target nodes. By leveraging CI, WSNs can achieve enhanced performance, reliability, and efficiency, addressing critical challenges and expanding their applicability across diverse real-world scenarios. While optimizing network operation and data accuracy, pinpointing moving target node locations remains difficult. This paper introduces a Hybrid DA-FA algorithm to address this challenge. A single anchor node, tasked with monitoring mobile targets, projects virtual anchor nodes in six directions, each 60 degrees apart, with equal range. When a mobile node enters the range of at least three virtual anchors, its 2D position is determined. Simulation results demonstrate the algorithm's effectiveness in localization efficiency and precision. Furthermore, the concept of virtual anchor nodes effectively mitigates line-of-sight limitations.

**Dragonfly Firefly (DA-FA) algorithm**

DA: The Dragonfly Algorithm (DA) draws inspiration from the swarming behavior of dragonflies, observed during hunting and migration. Hunting, or static swarm behavior, involves small groups making localized, sudden movements. Conversely, migratory, or dynamic swarm behavior, sees large swarms traveling long distances in unison. These behaviors highlight DA's exploration and exploitation capabilities.

DA's core principles include food attraction, alignment, cohesion, separation, and enemy avoidance. Each dragonfly in the swarm represents a potential solution within the search space. The algorithm utilizes five operators to govern dragonfly movement: separation (avoiding collisions), alignment (matching velocity), cohesion (moving towards the group center), attraction to food (seeking optimal solutions), and diversion from enemies (avoiding suboptimal areas). These operators dynamically adjust dragonfly positions, facilitating efficient search and optimization.

**Firefly algorithm (FA):** The Firefly Algorithm (FA), developed by Yang, utilizes swarm intelligence, mimicking fireflies' flashing behavior for attraction. Fireflies flash to signal and attract other fireflies, demonstrating activities like communication and predator warning. Yang's algorithm assumes fireflies are unisexual, attracting each other based on brightness: brighter fireflies attract dimmer ones. Brightness correlates with attractiveness and diminishes with distance. If no brighter fireflies exist, they move randomly. A firefly's brightness is determined by its fitness function's value. FA operates with parameters like attractiveness, absorption (light intensity reduction), and randomization (random movement), enabling exploration and optimization.

**Proposed hybrid DA-FA:** Effective optimization algorithms require a delicate balance between exploration and exploitation to function optimally. Exploration, or diversification, involves a broad search across the entire solution space, aiming to discover promising regions. Conversely, exploitation, or intensification, focuses on refining the best-known solutions within a localized area, seeking to maximize their potential.

An imbalance between these two crucial aspects can significantly degrade algorithm performance. Overemphasis on exploration may lead to inefficient searching and delayed convergence, while excessive exploitation risks premature convergence to local optima, hindering the discovery of truly optimal solutions. Therefore, a well-designed algorithm must strategically manage exploration and exploitation to ensure both comprehensive search and precise refinement, ultimately achieving superior results.

**The standard Dragonfly Algorithm (DA)** initiates its search by generating a random population of dragonflies, utilizing Levy flight to explore the solution space. This approach, combining random flight and Levy distribution, enhances the algorithm's exploration capabilities by expanding the range of potential solutions. Furthermore, DA effectively balances local and global search through adaptive tuning of its swarming parameters. This flexibility allows for dynamic adjustment, ensuring a thorough exploration of the search landscape.

While DA offers advantages like accurate approximations, it also presents challenges, notably slow convergence. This can limit its efficiency in time-sensitive applications. Similarly, the Firefly Algorithm (FA), despite its strengths, exhibits a tendency towards restrictive convergence rates. This limitation can hinder its ability to quickly locate optimal solutions, especially in complex search spaces.

Therefore, while both DA and FA are valuable optimization tools, their inherent limitations necessitate careful consideration. The need for faster convergence and improved exploration-exploitation balance highlights the potential benefits of hybrid approaches or modifications that address these shortcomings. By combining the strengths of different algorithms or refining existing ones, researchers aim to overcome these limitations and achieve more efficient and effective optimization.

To address the convergence limitations of both the Dragonfly Algorithm (DA) and the Firefly Algorithm (FA), a hybrid approach is proposed, combining their strengths for enhanced optimization performance. This hybrid DA-FA algorithm aims to expedite convergence by integrating FA principles into the DA framework.

Specifically, when a dragonfly fails to influence its neighboring elements, the Levy flight mechanism within DA is adjusted. This modification ensures that even in the absence of local influence, the algorithm continues to explore the search space effectively. Furthermore, the position update mechanism from FA, is incorporated into the DA-FA algorithm. This addition allows for more agile and responsive movement of the search agents, directly addressing the slow convergence issues prevalent in both individual algorithms.



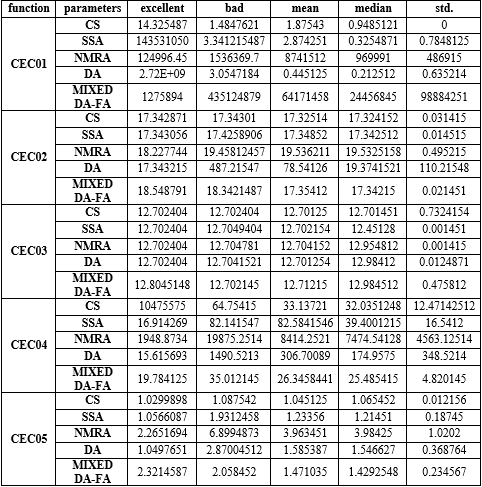
**FigURE2**,2D Localization Process

By strategically blending these two algorithms, the hybrid approach seeks to leverage the broad exploration capabilities of DA with the rapid convergence potential of FA. This synergy aims to provide a more efficient and effective optimization solution, particularly in scenarios where swift and accurate convergence is critical. The flowchart in Figure. 2 visually represents the 2D localization process using this proposed hybrid algorithm.

**Statistical Testing of Dragonfly-firefly (DA-FA) Algorithm**

The performance of DA-FA algorithms is assessed using the 05 challenging CEC 2019 '100-Digit Challenge' benchmark functions.

**TABLE 1** Results for CEC 2019 Benchmark functions



# **Results and Discussions**

Achieving optimal solutions in global optimization necessitates a balanced interplay between exploration and exploitation. Exploration, also known as diversification, involves a broad, global search across the entire solution space, aiming to identify promising areas. Conversely, exploitation, or intensification, focuses on refining the best-known solutions within a local region, leveraging current knowledge to enhance their quality.

An imbalance between these two crucial aspects can significantly hinder an algorithm's efficiency. Overly aggressive exploration can lead to dispersed searches, delaying convergence and potentially missing optimal solutions within promising local regions. Conversely, excessive exploitation risks premature convergence to local optima, preventing the discovery of superior global solutions.

Therefore, a well-designed optimization algorithm must strategically manage the trade-off between exploration and exploitation. Effective algorithms dynamically adjust their search strategies, ensuring both comprehensive exploration of the search space and focused refinement of promising solutions. This adaptive approach minimizes the risk of getting trapped in local optima and optimizes the convergence time, ultimately leading to more robust and accurate global solutions.

## **Statistical Testing of Dragonfly-firefly (DA-FA) Algorithm**

First, to assess the performance of the hybrid DA-FA algorithm, ten benchmark functions from the CEC 2019 "100-Digit Challenge" were employed. These functions, detailed in Table I, are standard tools for evaluating optimization algorithms.

The DA-FA algorithm was compared against other optimization algorithms, with each tested over 500 iterations using 30 agents. Table I demonstrates that DA-FA generally outperformed the other algorithms, with exceptions in CEC01 and CEC05.

This table further confirms that DA-FA achieved superior performance in more than half of the test cases, indicating its effectiveness compared to the other meta-heuristic algorithms.

Accurate localization in Wireless Sensor Networks (WSNs) presents a significant challenge, particularly when relying on a limited number of anchor nodes. To address this, a novel approach utilizing a single anchor node with virtual projections is proposed. This anchor node strategically projects virtual anchor nodes at six distinct angles within its coverage area. When a target node enters the range of this anchor node, two virtual anchor nodes and the original anchor node are selected, fulfilling the requirement of at least three nodes for 2D position determination.

The localization error, a crucial metric for evaluating accuracy, is computed using various met heuristic algorithms. However, the hybrid Dragonfly-Firefly Algorithm (DA-FA) has demonstrated superior performance, consistently achieving lower localization error values compared to other methods. This enhanced accuracy makes the proposed method highly suitable for a diverse range of applications, including animal tracking, supply chain management, coal mine personnel monitoring, and various industrial activities.

Moreover, the proposed approach contributes to energy efficiency within the network. By optimizing node localization and reducing unnecessary communication, it helps ensure that nodes conserve their energy, thereby extending their operational lifespan. This is particularly vital in WSNs where nodes often operate on limited battery power.

Looking ahead, the development of hybrid meta-heuristic algorithms holds immense potential for further improvements in localization accuracy and convergence time. Future research can focus on refining these algorithms, exploring new combinations of meta-heuristic techniques, and adapting them to address the specific challenges posed by different WSN applications. By continually advancing localization techniques, we can unlock the full potential of WSNs and enable their deployment in an even wider array of real-world scenarios.

# **Conclusion and Future Scope**

This paper comprehensively surveys Wireless Sensor Networks (WSNs), detailing their hardware, characteristics, and diverse real-world applications. It also addresses critical research challenges, particularly focusing on localization, a fundamental aspect of WSN functionality.

The paper provides a thorough overview of WSN localization, including evaluation criteria and performance metrics. It categorizes localization algorithms based on design and implementation strategies, reviewing techniques suitable for both static and dynamic WSN deployments.

A significant portion of the paper is dedicated to exploring Computational Intelligence (CI) based localization methods. These techniques, crucial for accurate node location estimation in multidimensional spaces, offer adaptable solutions for complex and challenging WSN environments. The paper highlights the implementation of CI algorithms in various WSN scenarios, demonstrating their effectiveness in optimizing node localization.

Furthermore, the paper introduces a hybrid Dragonfly-Firefly Algorithm (DA-FA) for enhancing localization accuracy, particularly for mobile target nodes. This algorithm, utilizing virtual anchor nodes, demonstrates improved performance compared to traditional methods.

Finally, the paper outlines future research directions, emphasizing the potential for further advancements in hybrid meta-heuristic algorithms to enhance accuracy, convergence time, and energy efficiency. These improvements will further expand the applicability of WSNs in diverse fields.

This paper introduces a novel Hybrid DA-FA algorithm, demonstrating its effectiveness against other meta-heuristics for range-based node localization in distributed, non-collaborative WSNs. Utilizing a single, centrally placed anchor node and introducing virtual anchor nodes, the algorithm efficiently tracks randomly moving target nodes within the designated area.

The proposed approach significantly improves localization accuracy and accelerates convergence, even with a reduced number of anchor nodes in mobile environments. This enhancement highlights the algorithm's suitability for various applications requiring precise and efficient node tracking. The introduction of virtual anchor nodes effectively addresses localization challenges, particularly in scenarios with limited anchor node deployment.

. Finally, a crucial future direction involves deploying these algorithms in real-world applications. Validating their performance in scenarios like coal-mine worker tracking or animal tracking in forests would provide invaluable insights and demonstrate their practical utility. Such applications would test the algorithms' robustness and adaptability in challenging and dynamic environments, bridging the gap between simulation and real-world implementation.

##### **References**

1. X. Zhang, Arizona State University, 1-20, 2016.
2. N. Saeed and B. R. Stojkoska, International Journal of Ad Hoc and Ubiquitous Computing. 23, 82-91 (2016).
3. L. Chelouah, F. Semchedine and L. Bouallouche-Medjkoune , Computers & Electrical Engineering., 1, 20 (2017).
4. Parulpreet, S Arun, Anil, K, Mamta, K A survey. Int. J. Curr. Eng. Sci. Res. 10, 74–80 ( 2017\_.
5. Singh, P, Khosla, A Kumar, AEU Int. J. Electron. Commun. 82, 543– 552 (2018)
6. G Sharma, A Kumar, Telecommunication Systems 67, 149–162 (2018).
7. R Salgotra, U.Singh, & amp; Saha, S. Expert Systems with Applications, 95, 384- 420 (2018).
8. T Nguyen. LN, Vy T. D, Y Shin, 1-20, (2019).
9. Brest, Janez Maucec, Mirjam Sepesy Boskovic, Borko, 19-26, (2019).
10. R. Salgotra, U. Singh, S. Saha, and A. H. Gandomi, IEEE Congress on Evolutionary Computation 1-20, (2020).
11. R Salgotra, U Singh, S Singh, G Singh, N Mittal, Applied Mathematical Modelling, 89, 188- 207 (2021).
12. W. Xia, Y. Jiang, X. Chen, R. Zhao,. 40, 609–624 (2022)
13. Sana Shahab, Mohd Anjum, M. Sarosh Umar, Int. J. Adv. Comput. Sci. Appl. 13 (3) (2022).
14. V. Ngunzi, F. Njoka, R. Kinyua, Heliyon, 9, 18823 (2023) .
15. J.A. Ternero, V. Mayor, R. Estepa, A. Estepa, G. Madinabeitia, Internet Things 24, 100981 (2023).
16. L.K. Ketshabetswe, AM. Zungeru, C.K. Lebekwe, B. Mtengi, Sci. African, 23, 2023 (2024).
17. Prakash J., L. Murali, N. Manikandan, et al., Sci. Rep. 14 , 468 (2024)
18. Z Y Huang, H T Zheng, Li C, Che C. Academic Journal of Science and Technology,  10, 33–39 (2024).
19. Y. Qiu, L. Ma, R. Priyadarshi , Arch. Comput. Methods Eng, 1—24 (2024).
20. D.M. John, S. Vincent, S. Pathan, K.M. Nayak, T. Ali, Results Eng. 23, 102412 (2024).
21. L.K.Ketshabetswe, A.M. Zungeru, C.K.Lebekwe, B.Mtengi, ResultsEng.23, 1026 (2024).