The Journal of Supercomputing (2025) 81:140
https://doi.org/10.1007/s11227-024-06616-6

EMGODV-Hop: an efficient range-free-based WSN node localization using an enhanced mountain gazelle optimizer

Reham R. Mostafa[1,2] · Fatma A. Hashim[3,4] · Ahmed M. Khedr[5] · Zaher AL Aghbari[5] · Imad Afyouni[5] · Ibrahim Kamel[6] · Naveed Ahmed[5]

Accepted: 14 October 2024

Abstract

Accurate node localization is essential in wireless sensor networks (WSNs) for effective data analysis and the successful operation of applications like environmental monitoring and disaster management. Range-free methods like the distance vector-hop (DV-hop) algorithm are often used due to hardware and cost constraints, but they face challenges in accuracy and stability. The NP-hard nature of the localization problem has led to the integration of metaheuristic algorithms in previous studies to enhance performance. This paper presents EMGODV-Hop, a novel approach for node localization in multi-hop networks that combines the DV-Hop algorithm with the mountain gazelle optimization (MGO) algorithm to enhance localization precision. The EMGODV-Hop method operates in two phases: First, it uses an improved variant of the DV-Hop algorithm to more accurately estimate distances between unknown and anchor nodes by incorporating a correction factor. Next, it employs an enhanced version of MGO algorithm, referred to as EMGO, to determine the positions of WSN nodes. The improved DV-Hop version enhances accuracy by incorporating a correction factor for better estimation of hop distances, while the EMGO algorithm addresses the limitations of the original MGO algorithm and improves its search capabilities. Extensive simulations assessed the effectiveness of the proposed method across various factors, including anchor node ratios, total node count, and communication ranges. The results demonstrate significant accuracy improvements with the proposed algorithm, showing enhancements of 48.69%, 26.22%, 19.33%, 28.21%, and 40.47% compared to DV-Hop, MGODV-Hop, PSODV-Hop, WSODV-Hop, and SSADV-Hop, respectively.

Keywords Wireless sensor networks (WSNs) · Node localization · Metaheuristics · DV-Hop algorithm · Mountain gazelle optimizer (MGO)

Extended author information available on the last page of the article

Published online: 06 November 2024

1 Introduction

Wireless sensor networks (WSNs) consist of small, cost-effective, battery-powered sensor nodes capable of wireless communication and self-organization, allowing them to operate effectively in challenging conditions [1]. These networks are used in applications such as disaster monitoring, wildfire surveillance, water quality assessment, and pollution detection [2, 3]. In some cases, large numbers of nodes must be deployed randomly in hostile environments, such as during natural disasters when sensors are dispersed from an aircraft without known locations. Installing GPS in each node is cost-prohibitive, and the lack of precise location data limits the utility of the sensor information. Accurate location information is crucial to interpret sensor data and implement location-sensitive data aggregation and routing algorithms [4], making the node localization process essential in WSN applications [5, 6].

Localization techniques in WSNs are classified into range-based and range-free algorithms [7]. Range-based algorithms estimate node positions using direct measurements like distance, angle, or signal strength, such as Radio Signal Strength Indicator (RSSI) [8], Time Difference of Arrival (TDoA) [9], and Angle of Arrival (AoA) [10]. Although these methods provide high accuracy, they require additional hardware and increase power consumption. In contrast, range-free algorithms, such as distance vector-hop (DV-hop) [11], centroid [12], amorphous [13], and approximate point in triangle (APIT) [14], rely on connectivity information and network topology without direct measurements. Range-free methods are generally more energy efficient and suitable for nodes with limited resources, although they may be less accurate.

DV-Hop algorithm is widely used in WSNs for localization due to its simplicity, low resource requirements, and adaptability to various network sizes, making it suitable for a wide range of WSN applications [15]. It estimates the positions of unknown nodes by calculating hop distances between anchor nodes (with known locations) and unknown nodes. The process involves gathering distance information based on hop counts among anchor nodes, estimating distances between anchors and unknown nodes, and computing positions using multilateration [16]. Although DV-Hop is easy to implement and energy efficient, its accuracy can be compromised by cumulative errors in hop distance estimation, which significantly affects the precision of position estimates through multilateration [17]. These errors can notably impact localization accuracy, particularly in large-scale networks.

Due to the complexity of node localization problems, it is formulated as an NP-hard optimization problem, leading to the adoption of various soft computing methods for its solution. For example, Cheng et al. [18] employed gradient techniques such as the Gauss–Newton algorithm for node localization. However, gradient-based techniques have the potential to be trapped in local minima and also require appropriate initial conditions for desired parameters. Annepu et al. [19] proposed the use of highly nonlinear artificial neural network (ANN) models, such as multilayer perceptron (MLP) and radial basis function (RBF), for nonlinear node localization tasks, emphasizing their effectiveness for stationary and mobile nodes.

As nonlinearity and constraints increase, traditional optimization techniques often struggle to find global optimal solutions. To address this challenge, the literature extensively explores the use of metaheuristic algorithms to improve node localization accuracy in WSNs. These algorithms, inspired by biological, physical, and social principles, are designed to tackle complex optimization problems effectively. For example, the genetic algorithm (GA) [20] simulates biological evolution, particle swarm optimization (PSO) [21] mimics the behavior of bird flocks or fish schools, and ant colony optimization (ACO) [22] emulates the food-searching behavior of ants. By incorporating these nature-inspired strategies, researchers have developed various approaches to enhance localization precision in WSNs. For example, studies like [23–26] focus on single-objective optimization algorithms, aiming to improve localization by optimizing a single criterion, such as accuracy. In contrast, works such as [27–31] employ multi-objective optimization models, which account for multiple factors like accuracy, energy efficiency, and communication overhead, offering a more balanced approach to node localization in WSNs.

Mountain gazelle optimizer (MGO) algorithm is a new metaheuristic algorithm inspired by the social life and hierarchy of wild mountain gazelles [32]. MGO incorporates four key components from the mountain gazelles' life cycle, including bachelor male herds, maternity herds, solitary territorial males, and movement for foraging. The developers of the MGO algorithm have underscored its competitive performance relative to established and contemporary metaheuristic algorithms. The algorithm's effectiveness has been rigorously validated through comprehensive statistical and convergence analyses, along with the Wilcoxon rank test, applied to a suite of 52 benchmark functions. In addition, the applicability of the MGO has been thoroughly evaluated in seven different engineering challenges. Experimental results show that MGO consistently outperforms or matches other leading algorithms in terms of accuracy, convergence speed, and robustness across various optimization problems. Moreover, due to its simplicity and efficiency, MGO has found successful applications in various domains, including medical feature selection [33], parameter extraction of photovoltaic models [34], neural networks [35, 36], and reactive power dispatch [37].

Although the MGO algorithm has demonstrated strong performance in addressing various optimization problems, it does have certain drawbacks. These include: (1) insufficient solution accuracy and slower convergence rates when applied to specific non-convex and high-dimensional numerical optimization tasks and (2) a tendency to stagnate in local optima, highlighting the need to enhance MGO's local and global search strategies for escaping local optima and improving overall solution quality.

This paper proposes a new localization method called EMGODV-Hop that is specifically designed for multi-hop networks. The technique combines two algorithms, the DV-Hop algorithm and the MGO algorithm, to improve the accuracy and efficiency of node localization in these networks. In phase one, an improved version of the DV-Hop algorithm estimates distances between anchor nodes and unknown nodes. In phase two, the enhanced variant of the MGO algorithm determines the precise positions of the WSN nodes. Through this integration of DV-Hop and MGO, the EMGODV-HOP technique aims to enhance node localization in multi-hop networks significantly. This research paper makes several significant contributions:

- An improved version of DV-Hop is proposed to calculate the mean hop distances of unknown nodes more accurately. This enhancement involves introducing a correction factor to optimize the hop distances by aligning them with the actual distances between the anchor nodes.
- An enhanced version of the MGO, named EMGO, is proposed that addresses certain shortcomings and enhances its search capabilities.
- A new localization method, EMGODV-Hop, is introduced, integrating improved DV-Hop for distance estimation and EMGO for estimating the coordinates of sensor nodes within WSNs.
- The simulation results demonstrate the superior performance of the proposed algorithm, outperforming other methods in terms of localization accuracy, stability, and scalability in different scenarios.

The paper follows a clear structure, starting with an exploration of existing issues and relevant research in Sect. 2. The original DV-Hop algorithm and the MGO algorithm are introduced in Sect. 3. The EMGODV-Hop algorithm is described in Sect. 4 in detail. The performance assessment and experimental results of the EMGODV-Hop algorithm are discussed in Sect. 5. Finally, Sect. 6 provides a concise summary of the study's findings, covering challenges, methodologies, algorithm details, performance evaluation results, and concluding insights.

2 Literature review

Substantial advancements have been made in the realm of range-free localization techniques for WSNs. DV-Hop algorithm, a notable example in this category, is noted for its simplicity, robustness, and adaptability across various network sizes. Numerous studies have aimed to enhance the accuracy of the original DV-Hop algorithm, which generally achieves an accuracy level of approximately 60–70%. These studies approach localization as an optimization problem, employing various metaheuristic algorithms to refine performance. This section concisely reviews the integration of metaheuristic algorithms with the DV-Hop method, highlighting the unique characteristics of our proposed EMGODV-Hop method and distinguishing it from other advanced algorithms. Table 1 provides a summary of recent research papers on the DV-Hop localization algorithm.

Cui et al. [25] proposed a new approach to improve DV-Hop performance for localization in cyber-physical systems. The approach utilizes an oriented cuckoo search (OCS) algorithm to optimize anchor placement and hop distance estimation in the DV-Hop algorithm. The authors compared the OCS-DV-Hop algorithm with the other methods, showing that OCS-DV-Hop achieves better localization accuracy, robustness, reduced computational complexity, and communication overhead. However, these studies focused on location performance in large areas while neglecting positioning performance in complex terrain. Mehrabi et al. [38] introduced the

Table 1 Recent approaches of utilizing metaheuristics to boost the DV-Hop performance

Author and year	Proposed method	Main strategies used		Merits	Practical Implications	Limitations
		Distance estimation	Position estimation			
Cui et al. [25]	OCS-DV-Hop	Same as Phase 1 and 2 in DV-Hop	Oriented cuckoo search algorithm (OCS)	Boosts the global search of CS using hybrid Levy-Cauchy distribution.	Improved precision of DV-Hop. - Enhanced location estimation for WSN	Increased algorithm complexity. - Incorrect calculation in hop distances
Mehrabi et al. [38]	SFLADV-Hop	Same as Phase 1 and 2 in DV-Hop	Genetic Algorithm (GA), Shuffled Frog Leaping (SFLA), and PSO	SFLA used to improve HopSize error. - Algorithm is straightforward and easy to apply.	Decreases the localization error compared to other methods.	Poor convergence. -trapping in local optima. - The localization accuracy improvement is unclear.
Cui et al. [39]	DECHDV-Hop	Change from discrete to continuous hop count	Differential Evolution (DE)	Adjust the hop count to ensure accuracy.	DECHDV-Hop achieves higher localization accuracy. - It reduces the error caused by estimated distance	Increased algorithm calculation overhead
Liu et al. [40]	HDCDV-Hop	Multi communication radius for hop number correction	Improved DE (IDE)	Boost the performance of DE algorithm to escape from local optima. - Correct the hop number to match reality.	Improved accuracy and stability of DV-hop algorithm. - better localization in random distribution of nodes	increased in computational complexity
Song et al. [41]	MGDV-Hop	Same as Phase 1 and 2 in DV-Hop	Enhanced GSO (MC-GSO)	Boost the search capability of GSO algorithm	Reduces the average location error and increases the location coverage	Poor stability, and increased algorithm complexity

Table 1 (continued)

Author and year	Proposed method	Main strategies used		Merits	Practical Implications	Limitations
		Distance estimation	Position estimation			
Chai et al. [24]	PWOADV-Hop	Phase 1 and 2 of DV-Hop	Parallel WOA (PWOA)	Enhances global search ability and population diversity of WOA	Proposed algorithm improves localization accuracy in WSN	Increased in computational complexity
Ghafour et al. [23]	SSIDV-Hop	Enhanced variant of DV-Hop	Squirrel search algorithm (SSA)	Increase accuracy by calculating the mean hop distance received from all anchors by nodes.	The algorithm has high accuracy, fast convergence rate, and good stability	Increased algorithm complexity - Trapping in local optima due to the use of SSA
Li et al. [42]	PCCSO-DV-Hop	Same as Phase 1 and 2 in DV-Hop	Parallel Compact Cat Swarm Optimization (PCCSO)	Three separate communication strategies used to improve local search of CSO	PCCSO-DV-Hop improves localization accuracy	Poor convergency - Larger memory consumption. -DV-Hop cannot indicate true distance
Ou et al. [43]	ICS-FGDV-Hop	Same as Phase 1 and 2 in DV-Hop	Improved cuckoo search with fuzzy logic and Gauss-Cauchy strategy (ICS-FG)	Tackles the main limitations of CS algorithm and boost its capabilities	Achieves lower positioning error than pcCS, IAGA, and others.	There is a need to enhance the stability and search capabilities of the proposed method.
Jia et al. [44]	CAFOA-DV-Hop	Same as Phase 1 and 2 in DV-Hop	Chaotic Adaptive Fruit Fly Optimization Algorithm (CAFOA)	High convergence speed. - Escaping local optima	Achieves higher localization accuracy than other methods	DV-Hop cannot indicate true distance. - High computational complexity

Table 1 (continued)

Author and year	Proposed method	Main strategies used		Merits	Practical Implications	Limitations
		Distance estimation	Position estimation			
Cao and Xu [17] [45]	OANSDV-Hop	Optimum anchor nodes subsets (OANS)	Continuous PSO	OANS used to correct the average hop distance. - BPSO is used to select optimum anchor node subsets	Improved localization accuracy in WSNs.	High Computational Complexity and energy consumption
Sun et al. [46]	2DHYP-GADV-Hop	Same as Phase 1 and 2 in DV-Hop	2D hyperbolic algorithm and an improved adaptive GA (IAGA)	Radio irregularity model shows algorithm effectiveness in anisotropic network.	The algorithm is adaptable and scalable. It improves the localization accuracy	Significant computational complexity arises from employing the hyperbolic method and IAGA.
Peng et al. [48]	PDM-TSMA	Proximity distance map (PDM)	Adaptive chaotic slime mold (TSMA)	TSMA enhances diversity and combines global/local search using adaptive chaos	Achieved higher localization accuracy and faster convergence speed compared to other algorithms	High computational complexity.

SFLADV-Hop algorithm, simplifying the identification of a feasible region for the unknown node's location. Enhancements in the average hop distance approximation are made using the Shuffled Frog Leaping Algorithm (SFLA), while position estimation employs a hybrid Genetic-PSO algorithm. Despite its outperformance of existing algorithms in localization accuracy and energy efficiency, the method faces increased complexity.

Cui et al. [39] proposed DECHDV-Hop to improve hop count accuracy in localization. By utilizing common one-hop nodes, DECHDV-Hop transformed discrete hop count values into continuous ones, resulting in precise location estimations and reduced errors. The Differential Evolution (DE) algorithm was used for the location estimation process. Evaluation of DECHDV-Hop demonstrated superior performance over other DV-Hop methods across four network simulation scenarios, particularly in C-shaped topologies where it reduced localization error by around 70%. While DECHDV-Hop boasts improved accuracy, it comes at the cost of increased computational time during processing. In another context, Liu et al. [40] presented HDCDV-Hop as a technique that combines the distributed DV-Hop algorithm with an enhanced DE method. This method divides the effective communication range of nodes into two segments by utilizing fractional hop counts and adjusting factors to enhance hop distance and minimize localization errors. The utilization of the improved DE method contributes to more precise localization results. However, one major drawback of this method is the increased computational complexity, as it requires conducting additional computations on each resource-constrained node.

Song et al. [41] proposed a hybrid strategy integrating glowworm swarm optimization (GSO) with DV-hop to enhance its accuracy and efficiency. The algorithm optimizes the selection of anchor nodes using the GWO algorithm while introducing a chaotic strategy to increase population diversity. Although it outperforms other algorithms in terms of localization accuracy, it may exhibit poor performance in certain cases due to being trapped in local optima rather than global optima. Chai et al. [24] proposed a parallelized version of Whale Optimization Algorithm (WOA) for improving the performance of DV-Hop in WSNs. The parallel approach optimizes anchor node placements and enhances information flow among multiple instances of PWOA, resulting in improved localization accuracy and faster convergence speed compared to traditional methods. Additionally, the use of communication techniques reduces overhead and improves scalability, although challenges may arise in large-scale networks due to memory limitations on individual nodes.

Ghafour et al. [23] introduced SSIDV-Hop by improving the estimation of the mean hop distance for unknown nodes to enhance accuracy. Unlike traditional DV-Hop, SSIDV-Hop calculates the mean hop distance based on the improved values from neighboring anchor nodes, resulting in increased precision. The localization process is carried out directly using the Squirrel Search Algorithm (SSA), eliminating the use of least square method. SSA incorporates a fitness function that considers weighted squared error between anchor and unknown node distances. Simulation results indicate that SSIDV-Hop surpasses other algorithms in terms of accuracy, stability, and convergence; however, it does come with a significant increase in computational overhead. Li et al. [42] introduced the Parallel Compact Cat Swarm Optimization (PCCSO) algorithm tailored for optimizing anchor node placement in the

DV-Hop method. This algorithm mitigates observed limitations in traditional CSO algorithms, such as slow convergence and susceptibility to local optima. The authors conducted a comparative analysis between PCCSO-DV-Hop and other methods. Simulation results suggest that PCCSO-DV-Hop achieves improved localization accuracy while reducing computational complexity and minimizing communication overhead. However, it is essential to note that CSO algorithm still faces challenges related to convergence rates and resource utilization.

Ou et al. [43] introduced the Improved Cuckoo Search algorithm with Fuzzy Logic and Gauss-Cauchy strategy (ICS-FG) to minimize DV-Hop localization errors in by integrating fuzzy logic and the Gauss–Cauchy strategy. This algorithm dynamically adjusts parameters using population diversity-based fuzzy logic and enhances search accuracy through the Gauss–Cauchy strategy. Experimental results demonstrate the superior performance of the ICS-FG approach in reducing positioning errors, although there is potential for further enhancement in stability and search capability. Jia et al. [44] proposed CAFOA-DV-Hop, an algorithm that combines the DV-Hop method with adaptive step variation chaotic fruit fly optimization algorithm (CAFOA). This algorithm uses an adaptive search step size strategy to improve global search ability and find a balance between global and local optimization. A unique chaotic strategy helps in faster convergence by avoiding local optimal solutions. Extensive simulations show that CAFOA-DV-Hop achieves high accuracy and fast convergence compared to other methods, although it has increased complexity and computational overhead as drawbacks.

Cao and Xu [45] proposed OANSDV-Hop, which utilized optimum anchor node subsets obtained through a binary PSO algorithm (BPSO). This approach recalculated the hop distance of each anchor node using OANS and shared this information with nearby unknown nodes to improve localization accuracy. The fitness function based on OANS was further optimized using continuous PSO to enhance accuracy. Results showed that OANSDV-Hop outperformed DV-Hop and other methods across various network settings. However, it comes with the drawback of increasing node computational burden and energy consumption. Sun et al. [46] proposed 2DHYP-GADV-Hop that combined the 2D hyperbolic localization method with an improved adaptive GA (IAGA) to improve node coordinate estimation and localization accuracy. It also considered the radio irregularity model for performance evaluation in anisotropic networks. Simulation results showed that this algorithm outperformed other methods in terms of accuracy and stability. However, due to its use of the hyperbolic approach and IAGA for position estimation, 2DHYP-GADV-Hop is computationally intensive.

Tagne et al. [47] introduced an enhanced PSO for indoor localization issues in WSN. In this version, each particle utilizes tabu search to find its best local neighbor and improve convergence toward a better solution. Additionally, limit and performance checks are included in the algorithm to evolve with superior particles within the constraint analysis space around the initial trilateration-based solution. This approach, named FPSOTS, employs the received signal strength indicator method to assess inter-sensor distances. Peng et al. [48] introduced TSMA, an adaptive chaotic slime mold algorithm for WSN node localization. TSMA integrates adaptive chaos to enhance population diversity and combines global and local search capabilities using an adaptive chaotic oscillation factor. Simulation results demonstrate an average improvement in localization performance of 28% to 46% across three diverse environments.

Previous studies have aimed to improve node localization accuracy but faced challenges due to high computational and communication demands, as well as limited search capabilities in metaheuristic algorithms. Therefore, this paper presents a new localization method, called EMGODV-Hop, which incorporates several enhancements. First, it applies a correction factor based on real distances between anchor nodes to adjust the average hop distances, leading to more accurate hop distance estimations for unknown nodes. Second, it incorporates an improved metaheuristic algorithm with enhanced search capabilities to further improve localization accuracy.

3 Background

This section introduces two algorithms: the traditional DV-Hop and MGO algorithms.

3.1 DV-Hop algorithm

DV-Hop is a widely used localization algorithm in WSNs that estimates node positions without relying on direct distance measurements. As a range-free method, it leverages connectivity information and distance vector routing to estimate the positions of unknown nodes using anchor nodes with known coordinates. DV-Hop approximates distances by multiplying the average hop distance from an anchor node by the hop count and then applies the least squares method (LSM) to determine the node positions relative to the anchors [11]. Due to its scalability and distributed operation, DV-Hop is particularly suitable for large-scale sensor networks. The algorithm operates in three main steps:

- Step 1 (Objective: Obtaining hop counts): Anchor nodes broadcast beacon packets containing their positions and an initial hop count set to zero. Upon receiving a beacon, each node checks its packet table. If the anchor node is not already recorded, the node adds the anchor's position and hop count to its table. If the anchor is already recorded, but the received hop count is lower, the node updates its table with the new hop count. This process of message forwarding and table updating continues, allowing nodes to determine the minimum hop count to each anchor node.
- Step 2 (Objective: Computation of average hop distance): In the second step, each anchor node computes its average hop distance as follows:

$$HopDis_i = \frac{\sum ds_{ij}}{\sum hc_{ij}} = \frac{\sum_{j=1,j\neq i}^{m-1} \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}}{\sum_{j=1,j\neq i}^{m} hc_{ij}} \tag{1}$$

where m refers to the count of neighboring anchors for anchor i, while $hc_{i,j}$ signifies the minimum hop count between anchor i and j. Once the average hop distance is computed, each anchor node broadcasts this value throughout the WSN. Unknown nodes, upon receiving the broadcast, adopt the first received average hop distance as their own ($HopDis_u$). Using this value, an unknown node can then estimate its distance to every anchor node in the network with the following equation:

$$\hat{ds}_{ui} = HopDis_u \times hc_{ui} \tag{2}$$

where $\hat{ds}_{ui}$ and hc_{ui} represent the estimated distance and minimum hop count between an unknown node u and an anchor node i.

- Step 3 (Objective: position estimation): In this step, the computed distance values are used to estimate the positions of unknown nodes using the LSM in conjunction with trilateration or multilateration [11]. When an unknown node detects at least n anchor nodes ($n > 2$), it solves a set of nonlinear equations to determine its coordinates, x_u and y_u:

$$\begin{aligned} \left(x_u - x_1\right)^2 + \left(y_u - y_1\right)^2 &= ds_{u1}^2 \\ \left(x_u - x_2\right)^2 + \left(y_u - y_2\right)^2 &= ds_{u2}^2 \\ &\vdots \\ \left(x_u - x_n\right)^2 + \left(y_u - y_n\right)^2 &= ds_{un}^2. \end{aligned} \tag{3}$$

To linearize these nonlinear equations, the last equation is subtracted from each of the preceding $n-1$ equations, resulting in:

$$\begin{aligned} 2x_u\left(x_1 - x_n\right) + 2y_u\left(y_1 - y_n\right) &= x_1^2 + y_1^2 + ds_{u,n}^2 - ds_{u,1}^2 - x_n^2 - y_n^2 \\ 2x_u\left(x_2 - x_n\right) + 2y_u\left(y_2 - y_n\right) &= x_2^2 + y_2^2 + ds_{u,n}^2 - ds_{u,2}^2 - x_n^2 - y_n^2 \\ &\vdots \\ 2x_u\left(x_{n-1} - x_n\right) + 2y_u\left(y_{n-1} - y_n\right) &= x_{n-1}^2 + y_{n-1}^2 + ds_{u,n}^2 - ds_{u,n-1}^2 - x_n^2 - y_n^2. \end{aligned} \tag{4}$$

This system of linear equations can be expressed in matrix form as:

$$FX = D \tag{5}$$

where X represents the coordinates (x_u, y_u) of the unknown node to be estimated. D is a vector containing constants derived from the equations and including distances squared. F is a matrix of coefficients representing the positions of the anchor nodes. The matrices F, X, and D are given by Eqs. 6, 7, and 8, respectively.

$$F = \begin{bmatrix} 2\left(x_1 - x_n\right) & 2\left(y_1 - y_n\right) \\ 2\left(x_2 - x_n\right) & 2\left(y_2 - y_n\right) \\ \vdots & \vdots \\ 2\left(x_{n-1} - x_n\right) & 2\left(y_{n-1} - y_n\right) \end{bmatrix} \tag{6}$$

$$D = \begin{bmatrix} x_1^2 + y_1^2 + d_{u,n}^2 - d_{u,1}^2 - x_n^2 - y_n^2 \\ x_2^2 + y_2^2 + d_{u,n}^2 - d_{u,2}^2 - x_n^2 - y_n^2 \\ \vdots \\ x_{n-1}^2 + y_{n-1}^2 + d_{u,n}^2 - d_{u,n-1}^2 - x_n^2 - y_n^2 \end{bmatrix} \tag{7}$$

$$X = \begin{bmatrix} x_u \\ y_u \end{bmatrix} \tag{8}$$

The LSM is then applied to solve this inconsistent system of equations, resulting in the estimated position X^* of the unknown node:

$$X^* = \left(F^T F\right)^{-1} F^T D \tag{9}$$

where the superscript T denotes matrix transposition.

3.2 Mountain gazelle optimizer (MGO)

MGO is a metaheuristic algorithm inspired by the social structure and behavior of mountain gazelles in the wild [32]. MGO incorporates four key factors from gazelle life: bachelor male herds, maternity herds, territorial males, and migration for food search. It selects mature male gazelles in the herd territory as the best global solution, because bachelor herds cannot procreate or rule. The algorithm picks the lowest cost one-third of the population for modeling and categorizes the rest as gazelles in maternity herds. MGO balances exploration and exploitation through speed and position adjustments, avoidance of poor solutions, and searching for new solutions.

3.2.1 Territorial solitary males (*TSM*)

MGO algorithm emulates the territorial behavior of adult male gazelles, capable of defending themselves and establishing isolated, well-defended territories against younger males. This behavior is incorporated into the algorithm by assigning higher fitness values to solutions closer to the best one, thus promoting convergence to optimal solutions and discouraging the exploration of less promising regions. This behavior is mathematically modeled as follows:

$$TSM = male_{gazelle} - \left| \left(ri_1 \times BH - ri_2 \times X(t) \right) \times F \right| \times \mathrm{Cof}_r \tag{10}$$

where the best global solution is denoted by $male_{gazelle}$. The parameters ri_1 and ri_2 are random integers selected from 1 or 2. The young male herd coefficient vector, BH, is calculated using Eq. 11, and F is computed using Eq. 12. The coefficient vector Cof_r, randomly selected and updated in each iteration, is used to increase the search capability and calculated using Eq. 13.

$$BH = X_{ra} \times \lfloor r_1 \rfloor + M_{pr} \times \lceil r_2 \rceil, ra = \left\{ \left\lceil \frac{N}{3} \right\rceil \dots N \right\} \tag{11}$$

Equation 11 computes the coefficient vector for the young male herd, where X_{ra} represents a random solution within the ra interval and M_{pr} represents the average number of search agents that are randomly selected. Additionally, N denotes the total number of gazelles, while r_1 and r_2 are random values between 0 and 1.

$$F = N_1(D) \times \exp\left(2 - Iter \times \left(\frac{2}{MaxIter}\right)\right) \tag{12}$$

In Eq. 12, N_1 is a normal-distributed random number, and *exp* represents the exponential function. *MaxIter* denotes the total number of iterations, and *Iter* represents the current iteration.

$$Cof_i = \begin{cases} (a+1) + r_3, \\ a \times N_2(D), \\ r_4(D), \\ N_3(D) \times N_4(D)^2 \times \cos\left(\left(r_4 \times 2\right) \times N_3(D)\right), \end{cases} \tag{13}$$

Equation 13 calculates a using Eq. 14, and r_3, r_4, and *rand* are random values from 0 to 1. N_2, N_3, and N_4 are random numbers in the normal range and the dimensions of the problem. r_4 is also a problem-dimensional random number between 0 and 1.

$$a = -1 + Iter \times \left(\frac{-1}{MaxIter}\right) \tag{14}$$

3.2.2 Maternity herds (MH)

Maternity herds play a critical role in the life cycle of the mountain gazelle because they are responsible for producing robust male offspring. Male gazelles may also contribute to the process of delivering offspring, as well as engage in competition with other males in their attempts to possess females for mating purposes. This behavior is modeled as follows:

$$MH = \left(BH + Cof_{1,r}\right) + \left(ri_3 \times \ male_{gazelle} - ri_4 \times X_{rand}\right) \times Cof_{2,r} \tag{15}$$

where *BH* represents the vector indicating the impact factor of young males, computed using Eq. 11. $Cof_{1,r}$ and $Cof_{2,r}$ are randomly selected coefficient vectors, independently calculated using Eq. 13. ri_3 and ri_4 are integer values randomly set to either 1 or 2. Finally, X_{rand} is a randomly selected gazelle's position.

3.2.3 Bachelor male herds (BMH)

As male gazelles reach maturity, they assert dominance by establishing territories and competing with other males for control over females. This competitive nature sometimes leads to aggressive encounters. This behavior is modeled mathematically as follows:

$$BMH = (X(t) - D) + \left(ri_5 \times male_{gazelle} - ri_6 \times BH\right) \times Cof_r \tag{16}$$

where $X(t)$ is the gazelle's position in the current iteration and D is determined using Eq. 17. ri_5 and ri_6 are random integers from 1 or 2.

$$D = \left(|X(t)| + |\ male_{gazelle}\ |\right) \times \left(2 \times r_6 - 1\right) \tag{17}$$

where r_6 is also a random number between 0 and 1.

3.2.4 Migration to search for food (MSF)

Mountain gazelles have a persistent foraging behavior, often embarking on long journeys for sustenance and migration. On the other hand, these gazelles boast rapid running speeds and formidable leaping capabilities. This behavior is mathematically modeled as follows:

$$MSF = (ub - lb) \times r_7 + lb \tag{18}$$

where ub and lb are the problem's upper and lower boundaries. r_7 is a random integer between 0 and 1.

4 Proposed EMGODV-Hop algorithm

This section presents the proposed localization method EMGODV-Hop for multi-hop networks. The method involves two phases:

- The first phase estimates unknown anchor node distances using an improved DV-Hop algorithm.
- Based on the calculated distances from the first phase, an enhanced MGO algorithm (EMGO) is utilized to estimate WSN node positions.

This approach integrates the best features of both techniques to boost localization precision in multi-hop networks.

4.1 Phase 1: improvements in average hop distance calculation

DV-Hop algorithm is a widely used localization method known for its simplicity and broad applicability. However, it has limitations related to the accuracy of distance estimates between anchor nodes and unknown nodes, primarily due to errors in average hop distance estimation at the anchor nodes. To address this issue, an improved version of the DV-Hop algorithm has been proposed, which integrates a correction factor to adjust the hop distance of anchor nodes during the second phase of the algorithm [49]. This adjustment aims to improve the precision of the localization process by correcting variations in hop distance, thereby enhancing the overall performance of the DV-Hop algorithm. The calculation of this correction factor is outlined as follows:

- In the DV-Hop algorithm's initial phase, nodes determine the minimum hop counts to anchor nodes. The coordinates of anchor nodes allow the computation of distances between them, reflecting their actual physical distances as follows:

$$ds_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{19}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of anchor nodes i and j.

- Additionally, the estimated distance between two anchor nodes, denoted as $\hat{d}s_{ij}$, is computed by multiplying the hop count hc_{ij} connecting them by the hop distance of either anchor node as follows:

$$\hat{d}s_{ij} = HopDis_i \times hc_{ij} \tag{20}$$

where $HopDis_i$ is the average hop distance of anchor i.

- The distance error, denoted as d^e_{ij}, is defined as the difference between the estimated distance and the actual distance, and it is computed as follows:

$$d^e_{ij} = \left| ds_{ij} - \hat{d}s_{ij} \right| \tag{21}$$

- The distance error d^e_{ij} related to an anchor node i is used to calculate the correction factor δ_i for that anchor node, which can be expressed as:

$$\delta_i = \frac{\sum_{i \neq j} d^e_{ij}}{\sum_{i \neq j} hc_{ij}} \tag{22}$$

- The hop distances of anchor nodes are adjusted by applying a correction factor, resulting in an improved average hop distance $HopDis^l_i$. This adjustment entails adding the correction factor to the previously computed hop distance as follows:

$$HopDis^l_i = HopDis_i + \delta_i \tag{23}$$

The hop distance for the unknown node is computed by averaging the improved hop distance gains of all neighboring anchor nodes, providing an improved average hop distance for the unknown node.

$$HopDis^I_u = \frac{\sum_{i=1}^{n} HopDis^I_i}{n} \tag{24}$$

Then, calculate the estimated distance $\hat{d}s_{ui}$ between a specific anchor node (i) and an unknown node (u) as follows:

$$\hat{d}s_{ui} = HopDis^I_u \times hc_{ui} \tag{25}$$

4.2 Phase 2: enhanced mountain gazelle optimizer (EMGO)

This section provides the proposed EMGO algorithm. It discusses the shortcomings of the conventional MGO algorithm and presents the EMGO framework, which focuses on evolving a population of problem solutions.

4.2.1 Limitation of original MGO algorithm

MGO algorithm, inspired by the group dynamics of mountain gazelles, utilizes four optimization strategies. However, the initial version of this algorithm encountered challenges such as slow and premature convergence and difficulty escaping local optima due to the limited exploration and exploitation capabilities and its dependence on greedy search. To address these issues, an enhanced version of MGO, called EMGO, has been introduced. EMGO aims to effectively evolve the solution population, mitigating constraints and addressing the shortcomings of the original MGO algorithm.

4.2.2 Architecture of the proposed EMGO method

This section presents the EMGO algorithm, which aims to address the limitations of the original MGO method. These limitations include slow convergence, susceptibility to suboptimal solutions, and an imbalance between exploration and exploitation. Algorithm 1 provides the pseudocode for EMGO, while its flowchart is depicted in Fig. 1. The main enhancements in EMGO involve:

- Control randomization parameter: Randomization is important in metaheuristic algorithms to prevent stagnation and early convergence to local solutions. To control this randomization, EMGO uses the parameter α, which balances

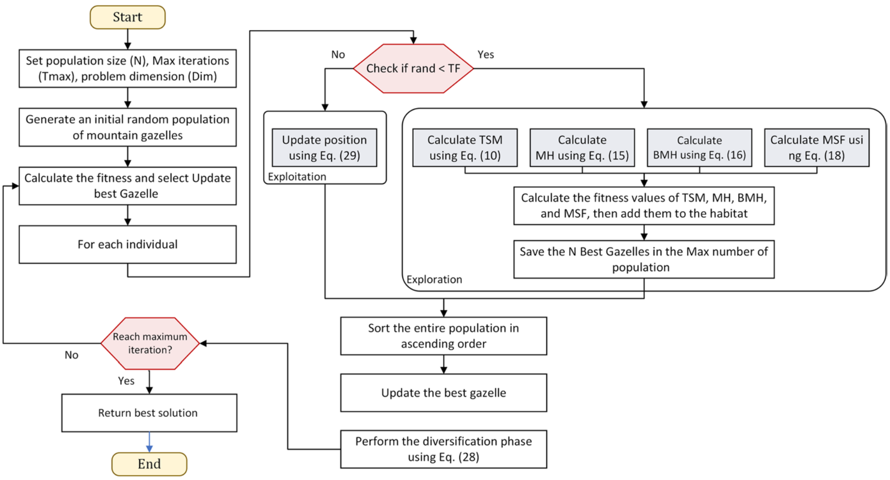


Fig. 1 Flowchart of the proposed EMGO algorithm

exploration and exploitation, reducing the chance of getting stuck in local optima. By generating both positive and negative random numbers, α allows a directional shift in the search process. This capability helps avoid getting trapped at suboptimal solutions and avoids premature convergence. The parameter α is defined as follows:

$$\alpha = 2 \times rand - 1 \tag{26}$$

- Transition factor (TF) parameter: The original MGO method lacks a transition parameter for a smooth shift between the exploration and exploitation stages, leading to unstable search behavior. To address this issue, a transition factor (TF) is introduced in the proposed EMGO algorithm. TF serves as a time-based balance, gradually shifting agents from exploration to exploitation as time passes. It can be expressed as follows:

$$TF = \exp\left(-\frac{Iter}{MaxIter}\right) \tag{27}$$

 where $MaxIter$ represents the total number of iterations and $Iter$ represents the current iteration. TF starts at one and gradually decreases as the search progresses, facilitating a smooth transition of agents from exploration to exploitation. This enables EMGO algorithm to achieve a more stable and efficient search process, enhancing its performance in various optimization problems.
- Proposed diversification operator: In the proposed EMGO algorithm, a portion of the new population undergoes diversification, controlled by a probability parameter called the probability of diversification (pd). The diversification operator in EMGO follows five strategies: (1) updating positions based on boundary constraints to explore new regions, (2) updating positions relative to the best MGO in the population, (3) modifying positions using archive information and a randomly selected MGO, with a control parameter a_j that is initialized randomly and decreases with problem size by being multiplied by another random number in the range [0,1], (4) selecting from a random archive pool ($archiveMGO$), and (5) retaining the current MGO without changes. This diversification operator helps the algorithm escape local optima and avoid stagnation in suboptimal solutions. The proposed diversification operator, as shown in Eq. 28, aims to prevent the algorithm from becoming trapped in local solutions. By introducing these five cases, the EMGO algorithm can explore different regions of the solution space and discover more effective solutions.

$$newMGO_{i,j} = \begin{cases} L_j + r_1 \times \left(U_j - L_j\right) & pd < 0.3 \\ best_{Gazelle,j} & pd < 0.4 \\ \left(1 - a_j\right) \times archiveMGO_{k,j} + a_j \times randMGO_j & pd < 0.5 \\ \mathrm{archiveMGO}_{k,j} & pd < 0.8 \\ MGO_{i,j} & otherwise \end{cases} \tag{28}$$

 where L_i and U_i are the lower and upper bounds, respectively.

- New exploitation phase: The original MGO algorithm lacked an exploitation phase, relying solely on randomization as per Eq. 10 for position updates. This approach led to suboptimal performance due to an imbalance in exploration and exploitation. To enhance efficiency, a new method was introduced that involves generating a temporary agent for each individual in every iteration to increase the chances of exploitation. If the temporary agent outperforms the current one, it will take its place. The temporary solution is generated using the following equation:

$$temp = x_i(t) + \alpha \times TF \times \left|x_r(t) - x_i(t)\right| \tag{29}$$

  where $x_i(t)$ stands for the current individual and $x_r(t)$ represents a randomly selected agent.

The EMGO process can be summarized as follows:

1. Initialization: An initial population of solutions is generated, and key parameters, such as the randomization control parameter (α) and transition factor (TF), are set.
2. Iteration process: The algorithm iteratively improves the population over a predetermined number of iterations, focusing on:
   - Fitness evaluation: Calculating the fitness of each solution based on the objective function.
   - Exploration and exploitation: Dynamically balancing exploration (searching new areas) and exploitation (refining existing solutions) using the TF parameter. A random number determines whether the focus is on exploration or exploitation.
   - Diversification: Applying a diversification operator to introduce variability and prevent stagnation, aiding in escaping local optima.
3. Best solution identification: The algorithm continuously updates and tracks the best solution based on fitness values throughout the iterations.
4. Termination: The process concludes when the specified number of iterations is reached, returning the best solution and its corresponding fitness value as the final output.

Algorithm 1 EMGO algorithm.

```
Input: Population size (N), max iterations (MaxIter), dimension of problem (Dim), and objective function f()
Output: Optimal solution
Generate the initial population of gazelle positions X_i
Initiate random value of a_j
for iter = 1 : MaxIter do
    Update TF using Eq.27
    for i = 1 : N do
        Evaluate the position of each gazelle according to the objective function f()
        Specify best_Gazelle
    end for
    for i = 1 : N do
        if rand < TF then
            // Exploration phase //
            Use Eq. 10 to estimate TSM
            Use Eq. 15 to estimate MH
            Use Eq. 16 to estimate BMH
            Use Eq. 18 to estimate MSF
            Determine the fitness of TSM, MH, BMH, and MSF and add them to the environment.
        else
            // Exploitation phase //
            Update position using Eq. 29
        end if
        Evaluate the fitness value
        Sort the population ascendingly.
        Update best_Gazelle
    end for
    for i = 1 : N do // Diversification operator //
        for j = 1 : Dim do
            Perform the diversification phase using Eq. 28
        end for
    end for
    iter++
end for
Return best solution.
```

4.3 EMGODV-Hop-based localization steps

In this section, we introduce the proposed localization method, EMGDV-Hop, for multi-hop networks. The method consists of two phases: An improved version of the DV-Hop algorithm is used to estimate distances between unknown nodes and anchor nodes, and the EMGO algorithm is employed to estimate WSN node positions based on the estimated distances in phase one.

Traditional distance estimation method in node localization typically involves multiplying hop count values by anchor nodes' hop distances. However, this method often results in inaccurate distance estimates, especially as the hop count between the anchor and unknown nodes increases, leading to decreased localization accuracy.

To improve precision, an improved DV-Hop algorithm is introduced, aiming to reduce errors and enhance accuracy in node localization. In addition, the EMGO algorithm is applied to estimate the positions of unknown nodes, replacing traditional trilateration or multilateration methods. The EMGO algorithm minimizes the squared error of the computed distances, transforming the position estimation process into a minimization task. By minimizing an objective function, the algorithm effectively calculates the coordinates of unknown nodes, improving localization accuracy.

The squared error of the estimated distances is defined as the sum of the squared differences between the estimated distance ds_{ui} and the Euclidean distance between the unknown node u and the anchor node i, which can be expressed as follows:

$$f(x,y) = 1/n \sum_{i=1}^{n} \left(\sqrt{\left(x' - x_i\right)^2 - \left(y' - y_i\right)^2} - d_{ui} \right)^2 \tag{30}$$

Note: n is greater than 2, indicating at least three neighboring anchors within the communication range of the unknown node u. Figure 2 displays the EMGODV-Hop node localization flowchart, while Algorithm 2 provides the pseudocode for EMGODV-Hop. The EMGODV-Hop algorithm determines the coordinates of N target nodes through the following steps:

- Step 1: Initialization.

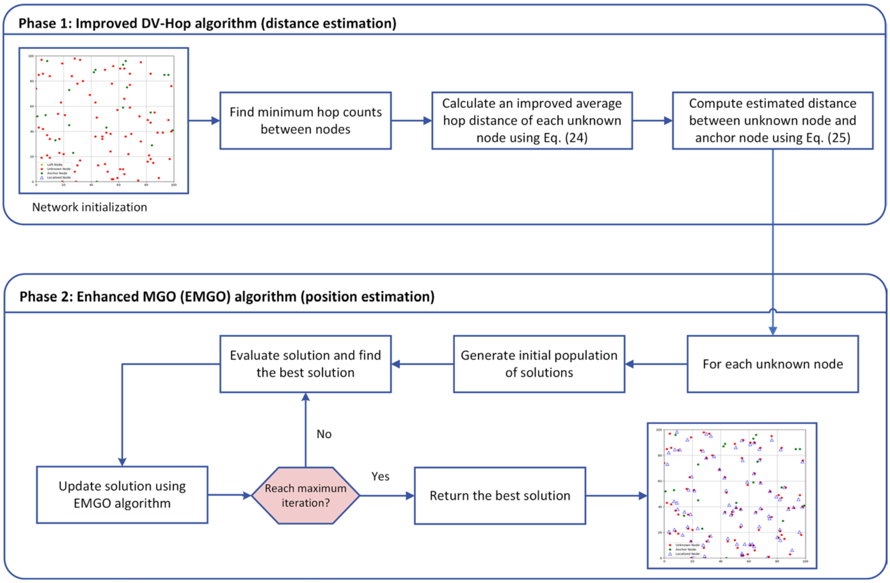


Fig. 2 Flowchart of the proposed EMGODV-Hop method

Initialize the parameters of the WSN, such as the dimension of the deployment area, communication range (R), and the total number of nodes.
Initialize the parameters of the metaheuristic algorithms: Population size (N), maximum iterations (*MaxIter*), problem dimension (D), upper bound (Ub), lower bound (Lb), and the objective function $f()$.

- Step 2: Network configuration. Configure the network with U unknown nodes and M anchors, where anchors may be equipped with GPS or deployed at known locations.
- Step 3: Estimate the distances between unknown nodes and anchors. This step is done as follows:

Anchor nodes send out an overwhelming number of beacon packets containing their coordinates.
Find the minimum hop counts between nodes in the entire network.
Compute anchor node average hop distance using Eq. 1.
Determine the actual and estimated distances between anchor nodes (Eqs. 19, 20).
Calculate the correction factor (δ_i) of each anchor node using Eq. 22.
Compute each anchor node's improved average hop distance $HopDis_i^I$ using Eq. 23.
Calculate the hop distance for the unknown node by averaging the improved average hop distance of the anchor nodes nearby using Eq. 24.
Use Eq. 25 to estimate an unknown node's distance from the anchor.

- Step 4: Position estimation using EMGO algorithm. This step is performed by each unknown node to localize itself by running the EMGO algorithm independently as follows:

Step 4.1: An initial random population is generated in a 2-dimensional deployment area using the formula $X = lb + r \times (ub - lb)$, where $r \in [0, 1]$, and lb and ub represent the lower and upper bounds, respectively.
Step 4.2: The fitness value is then calculated for every individual by considering its location as the coordinates of the unknown node using Eq. 30.
Step 4.3: The EMGO phases are performed to generate a new population that tries to minimize the fitness function.
Step 4.4: The previous procedures are repeated until the maximum number of iterations is reached. The location of the individual with the least fitness value is assumed to be the estimated location of the unknown node.
After getting localized, each unknown node starts acting as an anchor node and helps other localizable nodes get localized, indicating an increase in the number of anchor nodes as the iteration count progresses.
Repeat steps 4.1–4.5 until all the nodes become localized.

- Step 5: Evaluate the performance of EMGODV-Hop. The performance of the node localization process is analyzed in terms of ALE and localization ratio.

Algorithm 2 EMGODV-Hop algorithm

Input:
- Parameters of WSN
- Parameters of metaheuristic algorithms

Output: Unknown nodes' estimated locations
Place target and anchor nodes in the 2-D deployment area.
//// Distance estimation between unknown nodes and anchors ////
Calculates the average hop distance for the unknown node using Eq. 24
Determine an unknown node's distance from the anchor using Eq. 25
if unknown node calculates the distance to more than two anchors **then**
//// Position estimation using EMGO algorithm ////
  Initialize a random population of solutions $X = lb + r \times (ub - lb)$ where $r \in [0, 1]$
  Calculate each solution's fitness value using Eq. 30
  for $iter = 1 : MaxIter$ **do**
    Sort individuals by fitness value ascendingly and choose the best solution.
    Update TF using Eq.27
    for $i = 1 : N$ **do**
      if $rand < TF$ **then**
        // Exploration phase //
        Update positions using 10, 15, 16, and 18
      else
        // Exploitation phase //
        Update position using Eq. 29
      end if
      Evaluate the fitness value
      Sort the population ascendingly.
      Update $best_{Gazelle}$
    end for
    for $i = 1 : N$ **do** // Diversification phase //
      for $j = 1 : Dim$ **do**
        Perform the diversification phase using Eq. 28
      end for
    end for
    $k++$
  end for
  Return best solution.
end if

5 Simulation results and discussion

This paper presents two sets of experiments that illustrate the efficiency and robustness of the EMGO algorithm. The first set employs the algorithm as a global optimization tool to identify optimal values for the IEEE Congress on Evolutionary Computation 2020 (CEC'20) benchmark functions. The second set concentrates on evaluating the proposed localization algorithm, known as EMGODV-Hop, which is built upon EMGO, with the aim of assessing its capability to achieve highly precise node localization within WSNs. The experiments were carried out on a PC equipped with an Intel® CoreTM i7 3.40 GHz processor, 16GB of RAM, and Microsoft Windows 11. The experiments were carried out using MATLAB 2022b.

5.1 Experimental series 1: CEC'20

This experiment aims to assess the effectiveness of the EMGO algorithm in solving complex mathematical functions. The performance of EMGO is compared against other optimization algorithms using the CEC'20 test suite [50], which encompasses diverse types of mathematical functions, including unimodal (F1), multimodal (F2-F4), composition (F5-F7), and hybrid (F8-F10) functions designed specifically to evaluate its capacity for exploration and exploitation as well as its ability to escape local optima. Each benchmark function was evaluated with a dimension of 10 over 30 separate runs, with a maximum of 1000 iterations set for each run.

The performance of the algorithms was evaluated using the mean and standard deviation (STD) of fitness values. The experimental findings are presented in Table 2, where the bold values indicate the optimal solutions obtained. Furthermore, the final row illustrates the Friedman rank of each method. The results of the EMGO are compared with several state-of-the-art metaheuristic algorithms. These include Archimedes Optimization Algorithm (AOA) [51], Equilibrium Optimizer (EO) [52], Harris Hawks Optimization (HHO) [53], Dandelion Optimizer (DO) [54], PSO [21], WOA [55], Fick's Law Algorithm (FLA) [56], and Coati Optimization Algorithm (CoatiOA) [57]. All experiments used the default values specified in the original publications for the parameters of the comparing algorithms.

5.1.1 Exploration and exploitation evaluation

Unimodal test functions are a reliable benchmark for evaluating optimization algorithms, particularly in assessing their ability to effectively exploit the search space and identify the global optimum. Conversely, multimodal functions, characterized by multiple local minima, provide a more challenging evaluation, testing an algorithm's exploration capabilities and its effectiveness in avoiding entrapment in local optima. For the unimodal test function (F1), PSO demonstrates the best performance with a mean value of 1558.567, followed by MGO, which achieves a mean value of 2433.78. EMGO ranks fourth with a mean value of 3583.6, indicating its competitive ability to locate optimal solutions. Despite being surpassed by PSO and MGO, EMGO significantly outperforms many other algorithms, showcasing its strong capability to efficiently exploit the search space to approach the global optimum. This underscores EMGO's potential and strength in various problem domains requiring high exploitation capabilities.

EMGO shows a marked improvement on multimodal functions (F2-F4), demonstrating its superiority over other algorithms. For function F2, EMGO achieves the best mean value of 1356.55, reflecting its robust exploration capabilities, followed by EO and FLA with means of 1522.28 and 1609.6, respectively. The lowest performance is by CoatiOA at 2615. In function F3, EMGO again leads with the lowest mean value of 720.67, underscoring its effectiveness in navigating

Table 2 Comparison outcomes for each method utilizing the CEC'20 functions with $Dim = 10$

Function	Measures	EMGO	MGO	AOA	EO	HHO	DO	PSO	WOA	FLA	CoatiOA
F1	Mean	3583.649	2433.784	7.55E+09	3229.405	516412.9	5469.349	**1558.567**	8791697	88082.4	1.03E+10
	Std	2370.553	2263.273	2.37E+09	2355.094	310147.8	3469.904	1419.532	11279517	70549.36	4.52E+09
	Rank	4	2	9	3	7	5	1	8	6	10
F2	Mean	**1356.555**	1742.376	2329.025	1522.286	2040.596	1729.296	1703.593	2141.904	1609.608	2615.006
	Std	161.7	206.0862	198.524	218.808	230.3407	181.5269	349.246	389.8211	187.8558	**158.6266**
	Rank	1	6	9	2	7	5	4	8	3	10
F3	Mean	**720.6768**	729.5005	777.9986	723.5053	787.1355	752.0973	723.7334	780.7109	722.7378	801.5245
	Std	**5.151659**	8.087575	16.11462	6.839207	19.35643	17.7236	5.79222	20.06556	5.884219	20.50224
	Rank	1	5	7	3	9	6	4	8	2	10
F4	Mean	**1901.087**	1901.141	22954.04	1901.163	1907.271	1902.347	1901.097	1907.936	1901.916	40376.73
	Std	0.492562	0.458234	32715.31	**0.395737**	3.329932	1.127299	0.495009	6.007057	0.797457	49432.41
	Rank	1	3	9	4	7	6	2	8	5	10
F5	Mean	8060.927	5516.832	124433.9	6049.532	45056.37	7388.171	**3666.791**	149441.9	84531.2	327099.7
	Std	3384.99	3224.411	119035.3	3345.225	46737.83	3874.008	**2007.524**	262056.3	163718.7	214576.2
	Rank	5	2	8	3	6	4	1	9	7	10
F6	Mean	**1669.232**	1711.289	1845.313	1679.587	1846.55	1723.055	1738.508	1844.737	1771.222	2053.791
	Std	**67.41885**	77.0921	70.64044	84.2105	121.6	75.86468	100.0677	117.9298	125.1033	152.1565
	Rank	1	3	8	2	9	4	5	7	6	10
F7	Mean	3580.148	4788.2	355656	2565.075	10423.92	8892.371	3326.836	197367.1	229645.3	10189.6
	Std	788.0242	3916.181	644052.5	**259.5904**	11823.19	5365.273	604.4239	244618.9	557751.3	7508.243
	Rank	3	4	10	1	7	5	2	8	9	6
F8	Mean	**2296.156**	2301.212	2538.178	2300.643	2358.749	2382.761	2388.964	2384.236	2362.922	3111.675
	Std	0.429174	19.47857	183.6081	**0.389013**	280.8781	304.1356	300.3324	254.8454	236.6952	314.7273
	Rank	1	3	9	2	4	6	8	7	5	10

Table 2 (continued)

Function	Measures	EMGO	MGO	AOA	EO	HHO	DO	PSO	WOA	FLA	CoatiOA
F9	Mean	**2717.45**	2745.293	2818.653	2743.529	2810.936	2773.616	2738.636	2781.529	2741.242	2864.228
	Std	7.671325	9.204298	54.18072	**5.603475**	106.884	25.12392	45.72803	52.07235	66.42277	65.49058
	Rank	1	5	9	4	8	6	2	7	3	10
F10	Mean	**2922.632**	2925.904	3204.677	2929.953	2934.177	2934.869	2926.755	2945.381	2934.881	3475.555
	Std	**21.75897**	23.96157	138.7835	22.35251	23.65813	23.94756	23.17883	54.49113	22.13783	248.8001
	Rank	1	2	9	4	5	6	3	8	7	10
Friedman’s mean rank		1.9	3.5	8.7	2.8	6.9	5.3	3.2	7.8	5.3	9.6
Rank		1	4	9	2	7	5	3	8	6	10

The best results are highlighted in bold complex search spaces. FLA and EO rank closely with means of 722.73 and 723.5, while CoatiOA shows the weakest performance at 801.52. Similarly, for Function F4, EMGO secures the best mean value of 1901.08, slightly outperforming PSO and MGO, with means of 1901.08 and 1901.14, respectively. CoatiOA remains the least effective, with a significantly higher mean of 40376.7. EMGO's strong performance in multimodal functions highlights its effectiveness in tackling problems requiring extensive exploration across diverse regions of the solution space and identifying multiple optimal solutions.

5.1.2 Evaluation of local optima avoidance

Composition and hybrid functions are specifically designed to test an optimization algorithm's ability to effectively escape local optima. Applying EMGO to these functions highlights its ability to avoid being trapped in suboptimal solutions. Moreover, these functions provide benchmarks for evaluating the algorithm's balance between exploration and exploitation.

The results presented in Table 2 clearly demonstrate the superiority of EMGO in most hybrid functions (F5-F7). For Function F5, PSO achieves the best mean value of 3666.79, followed by MGO, EO, DO, and EMGO with means of 5516.8, 6049.5, 7388.1, and 8060.92, respectively. For Function F6, EMGO secures the best mean value of 1669.23, outperforming EO and MGO, with means of 1679.58 and 1711.28, respectively. CoatiOA remains the least effective, with a significantly higher mean of 2053.79. For Function F7, EO demonstrates the best performance with a mean value of 2565.07, followed by PSO and EMGO, which achieves a mean value of 3326.83 and 3580.14, respectively. Additionally, Table 2 displays the solutions obtained by EMGO and other algorithms when tackling composition functions (F9-F10), firmly establishing EMGO's dominance over its counterparts. These findings highlight the superior performance of EMGO, demonstrating an effective balance between exploration and exploitation. This balance enhances its ability to avoid local optima, establishing EMGO as a robust optimization algorithm well suited for navigating complex problem spaces.

5.1.3 Statistical analysis

The experimental evaluation demonstrates that the EMGO algorithm outperforms other comparable algorithms. To validate and confirm the rankings obtained from these experiments, Friedman's test and Wilcoxon's rank-sum tests were employed. These statistical tests were conducted to establish and substantiate the superior performance of EMGO relative to its counterparts.

Friedman test (F_{test}), as proposed by Derrac et al. [58], is a nonparametric statistical method used in this experiment to rank EMGO relative to other algorithms based on their achieved fitness values. The ranking procedure is conducted as follows:

$$F_{\text{test}} = \frac{12n}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{31}$$

where k represents the number of algorithms, R_j denotes the mean rank of the $j-th$ algorithm, and n signifies the number of case tests. This statistical test initially determines the rank of each algorithm independently and subsequently computes the average rank to derive the final rank for each algorithm concerning the specific problem under consideration. The results in Table 2 clearly indicate that the EMGO algorithm achieves the highest rank according to the Friedman test. This comprehensive ranking highlights the superior performance of the proposed EMGO algorithm compared to all other algorithms considered in the evaluation.

Wilcoxon's rank-sum test is a nonparametric statistical method used to compare the performance of competing algorithms. By calculating a p-value, this test allows the analysis of whether there is a statistically significant difference in performance between two algorithms. The results of the Wilcoxon test that compares EMGO with its competitors are presented in Table 3. The alternative hypothesis is accepted, as the majority of the p-values in the table are below 5%, indicating a significant difference between the performance of EMGO and the other algorithms. Thus, the findings and discussion in this study validate the effectiveness of the exploration and exploitation capabilities of EMGO.

5.1.4 Convergence performance analysis

Convergence curve is a graphical representation that depicts the algorithm's progression toward the optimal solution across successive iterations. Figure 3 illustrates a comparative analysis of convergence curves involving the EMGO algorithm and its counterparts for the CEC'20 test functions. This visualization enables an assessment of how rapidly and efficiently each algorithm converges to an optimal or near-optimal solution through iterative improvements.

The figure demonstrates the EMGO's capacity to rapidly approach near-optimal solutions, achieving this with high efficiency by requiring significantly fewer iterations. The convergence curves for EMGO consistently and rapidly decrease across various optimization stages, encompassing diverse test function families-unimodal, multimodal, composition, and hybrid functions. This visual evidence underscores EMGO's superiority over competing algorithms in terms of convergence speed and efficiency. This performance advantage is attributed to the algorithm's well-balanced exploration and exploitation strategies. EMGO's effective integration of these strategies enables it to navigate complex search spaces efficiently, avoid local optima, and converge quickly to high-quality solutions.

5.1.5 Boxplot behavior analysis

The boxplot is a visual aid that efficiently summarizes statistical data by displaying important metrics such as the median, quartiles, and possible outliers. It offers an overview of the distribution of numerical variables, providing valuable

Table 3 p value of the Wilcoxon test between EMGO and other algorithms

EMGO versus	MGO	AOA	EO	HHO	DO	PSO	WOA	FLA	CoatiOA
F1	0.662735	3.02E-11	0.8766349	3.02E-11	0.007959	0.1579757	3.02E-11	3.0199E-11	3.0199E-11
F2	5E-09	3.02E-11	0.0042259	7.389E-11	2.9215E-09	9.211E-05	3.474E-10	4.1178E-06	3.0199E-11
F3	3.83E-05	3.02E-11	0.1373228	3.02E-11	3.1589E-10	0.037782	3.02E-11	0.2009489	3.0199E-11
F4	0.599689	3.02E-11	0.3478278	3.338E-11	3.2555E-07	0.8766349	3.338E-11	1.3367E-05	3.0199E-11
F5	0.56922	2.03E-09	0.8187457	1.996E-05	0.33285469	0.0090688	5.092E-08	0.09925761	8.1527E-11
F6	0.003034	5.07E-10	0.2580515	1.01E-08	0.00090307	0.0069724	3.352E-08	0.00031821	3.6897E-11
F7	0.211561	6.7E-11	0.040595	5.607E-05	2.879E-06	0.8418015	1.957E-10	0.00037704	6.0459E-07
F8	0.001174	3.02E-11	0.0020016	1.067E-07	6.5261E-07	0.0156381	8.485E-09	2.6695E-09	3.0199E-11
F9	0.579294	8.35E-08	0.2115612	1.873E-07	4.1127E-07	0.7561685	5.967E-09	0.00117376	7.1186E-09
F10	0.01988	3.02E-11	0.0103147	0.5692202	0.00666888	0.0038461	0.0002006	0.43764134	3.0199E-11

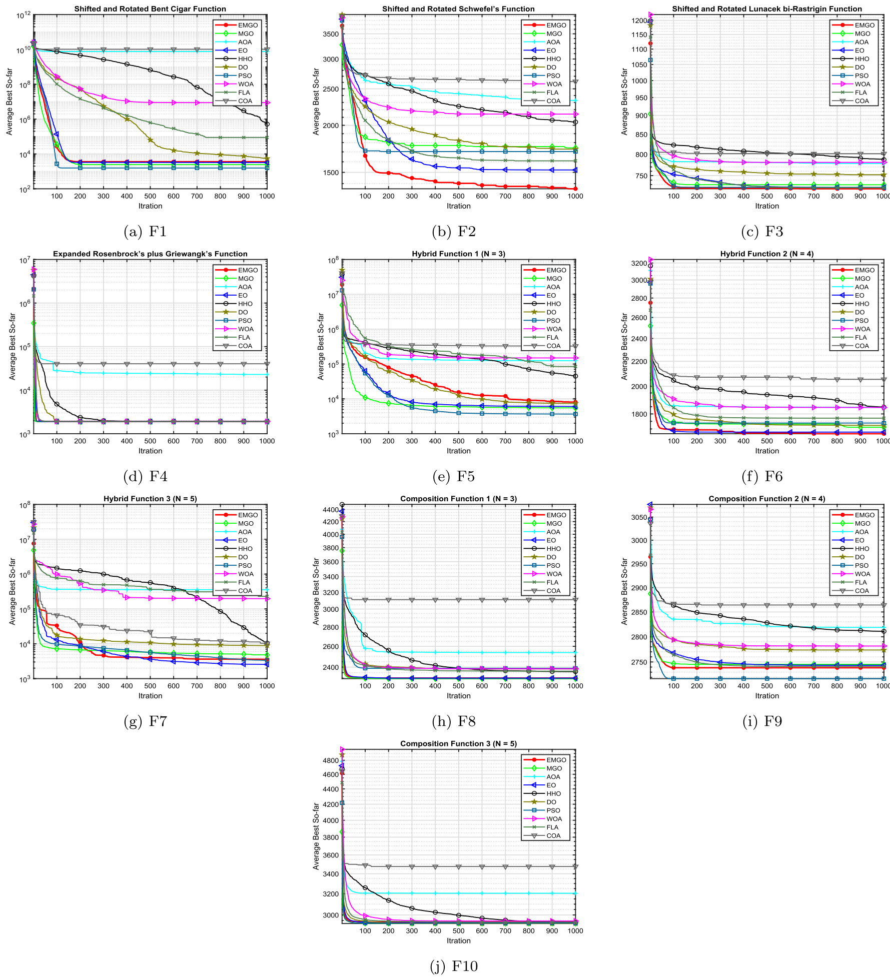
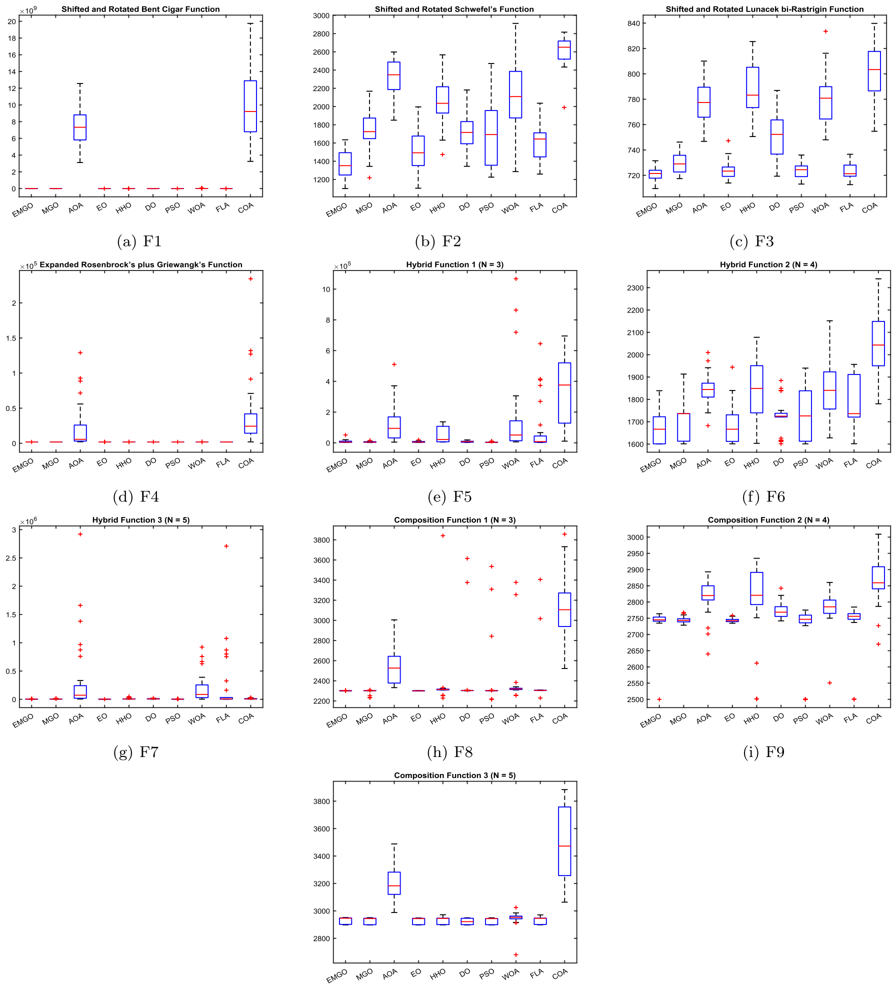


Fig. 3 Proposed EMGO and competing methods' convergence curves across the CEC'20 test suite at $Dim = 10$

insights into the data's spread and central tendency. Figure 4 depicts the boxplot of EMGO alongside other algorithms. It is evident that EMGO's boxplots exhibit notably lower values and a narrower range across the ten functions assessed for dim = 10. This feature signifies EMGO's superior performance compared to the other algorithms tested. The narrower range implies a higher level of consistency among data points, indicating EMGO's consistency in achieving the same results across various runs. Moreover, the lower values in EMGO's boxplots emphasize its ability to produce more optimal solutions when compared to its counterparts. These characteristics within EMGO's boxplot highlight its strong performance and consistent proficiency, illustrating its effectiveness in handling diverse functions or problems.



(a) F1 (b) F2 (c) F3 (d) F4 (e) F5 (f) F6 (g) F7 (h) F8 (i) F9 (j) F10

Fig. 4 Proposed EMGO and competing methods' boxplot curves across the CEC'20 test suite at $Dim = 10$

5.2 Experimental series 2: node localization in WSNs

Extensive simulations are performed in this section to evaluate the performance of the proposed EMGODV-Hop localization algorithm using various commonly used measures. Its effectiveness is validated through the comparison with the original DV-Hop algorithm under similar network conditions. Furthermore, various metaheuristics, including PSO [21], white shark optimizer (WSO) [59], salp swarm algorithm (SSA) [60], and original MGO algorithm, are selected and evaluated to ensure a fair comparison with the proposed EMGO. It is worth

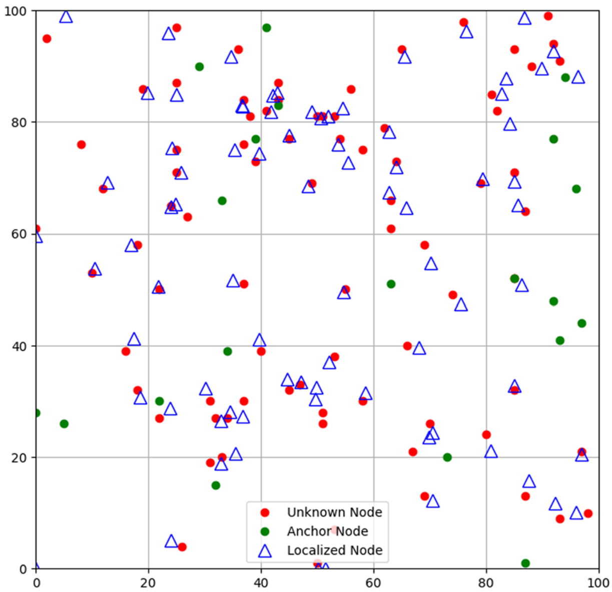


Fig. 5 WSN sensor node distribution

Table 4 WSN settings

Setting parameter	Value
Domain of networks	100×100 (m^2)
Node count	100–400
Total anchor nodes	20–60
Communication range	20–60 (m)

highlighting that all algorithms are evaluated under uniform conditions, utilizing 30 search agents and a fixed maximum of 50 iterations.

5.2.1 Simulation parameters

During the simulation, nodes are randomly distributed within a $100m \times 100m$ deployment area, as depicted in Fig. 5. The results presented are the average of 50 independent runs. The WSN settings are listed in Table 4. The parameters of the metaheuristic algorithms used in the comparison are set according to their original papers.

5.2.2 Performance measures

The performance of the proposed EMGODV-Hop algorithm is evaluated using two key metrics: average localization error (ALE) and localization success ratio. These measures are defined as follows:

- Average Localization Error (ALE) is a straightforward metric that quantifies the accuracy of the algorithm by measuring the average difference between the estimated and actual positions of the nodes. A lower ALE indicates higher precision in the node localization process. ALE is calculated as follows:

$$ALE = \frac{\sum_{u=1}^{N_u} \sqrt{\left(x_u^* - x_u\right)^2 + \left(y_u^* - y_u\right)^2}}{R \times N_u} \tag{32}$$

 The estimated location of an unknown node u is denoted by $\left(x_u^*, y_u^*\right)$, while $\left(x_u, y_u\right)$ represents its true location. N_u is the number of unknown nodes that can be localized, and R is the communication range.
- The localization success ratio metric represents the proportion of nodes successfully localized by the algorithm relative to the total number of nodes. A higher localization ratio signifies the algorithm's effectiveness in accurately determining the positions of a greater number of nodes.

The evaluation of all algorithms involves 50 independent runs of random WSN node distributions. Several statistical metrics are employed to gauge the effectiveness of the proposed algorithms.

1. *Mean*: To ensure an accurate evaluation, the algorithms are tested on 50 different and independent random distributions of WSN nodes ($M = 50$). The average of the ALE, as defined in Eq. 33, is calculated to obtain a reliable estimate of the localization accuracy.

$$Mean = \frac{1}{M} \sum_{i=1}^{M} f_*^i \tag{33}$$

 where f_*^i indicates the ALE value of the optimal solution generated at the $i - th$ run.
2. *Standard deviation (STD)* : The algorithm's stability and robustness can be evaluated with STD. Consistent convergence to the same solution by the optimization method, as measured by a small STD, is preferable to the more unpredictable behavior that can be expected from an approach with a large STD. The following equation is used to calculate the STD:

$$STD = \sqrt{\frac{1}{M-1} \sum \left(f_*^i - Mean\right)^2} \tag{34}$$

5.2.3 Influence of anchor ratio variation

The accuracy of the localization algorithm is significantly affected by the number of anchor nodes within the network. This section examines how variations in the anchor ratio influence localization accuracy. In this experiment, a total of 100 sensor nodes were randomly distributed within a sensing area of 100 $m \times 100m$, with each node having a communication range of 30 m. The experiment evaluated the impact of increasing the anchor ratio from 20% to 60%, in increments of 10%, on the localization error. The findings are summarized in Fig. 6 and Table 5.

Figure 6 presents the variation in ALE with a 97% confidence interval (CI), plotted against the number of anchor nodes. The CI serves as an indicator of the reliability and stability of the localization algorithm, with narrower intervals suggesting greater consistency and reliability in its performance. This analysis reveals the impact of varying the anchor ratios on the accuracy and stability of the localization method. As illustrated in Fig. 6, the accuracy of all algorithms in estimating positions improves with an increasing number of anchor nodes. As the number of anchor nodes increases, both the distance and hop count between them decrease, leading to a more accurate measurement of the average hop distance. This improved accuracy in the average hop distance reduces errors in estimating distances between unknown and anchor nodes, resulting in more precise estimates. These accurate distance estimations are crucial for enabling the metaheuristic algorithm to find optimal solutions to locate unknown nodes.

Table 5 displays the ALE of different algorithms, including the EMGODV-Hop algorithm, for varying anchor ratios. The results demonstrate that the EMGODV-Hop algorithm achieves a lower localization error rate compared to other methods. Specifically, as the anchor ratio ranges from 20% to 60%, the localization error decreases from 0.47 to 0.29 for EMGODV-Hop, from 0.55 to 0.44 for MGODV-Hop, from 0.51 to 0.37 for PSODV-Hop, from 0.54 to 0.46 for WSODV-HOP, 0.65 to 0.56 for SSADV-HOP, and from 0.73 to 0.65 for DV-hop algorithms. The EMGODV-Hop algorithm demonstrates the lowest STD values compared to other evaluated algorithms, indicating a high degree of consistency in its results across independent runs. In addition, Table 5 presents the average ALE values for various localization algorithms, along with their corresponding tolerance (TOL) levels. These tolerances are set to achieve 97% accuracy when comparing the average value to the specified tolerance. The EMGODV-Hop method demonstrates superior localization accuracy compared to other algorithms. On average, there is

Fig. 6 ALE of the comparative algorithm for different numbers of anchor nodes

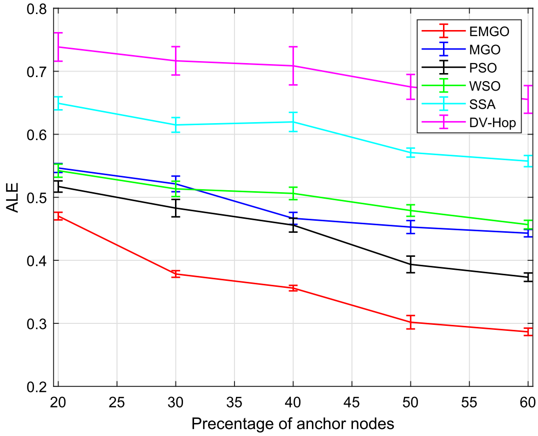


Table 5 Experiments comparing the comparative algorithm's performance with varying numbers of anchor nodes

# anchor nodes	Measures	EMGODV-Hop	MGODV-Hop	PSODV-Hop	WSODV-Hop	SSADV-Hop	DV-Hop
20%	Mean	0.470233048	0.546509679	0.517151879	0.54228575	0.64919539	0.73857571
	STD	0.017628891	0.018435506	0.023132154	0.02619932	0.02658919	0.05786665
30%	Mean	0.378346192	0.521291099	0.482875844	0.51339405	0.61494221	0.71664025
	STD	0.013915302	0.032267159	0.035511521	0.03160297	0.02984903	0.05779645
40%	Mean	0.355940032	0.466654902	0.455872446	0.5061944	0.61964222	0.70869193
	STD	0.013161807	0.024545523	0.027352392	0.02524704	0.03897232	0.03081241
50%	Mean	0.301802063	0.452815761	0.393545318	0.47904978	0.57098216	0.67527725
	STD	0.026678186	0.026377426	0.033638388	0.02357397	0.01835071	0.05057338
60%	Mean	0.286579628	0.443105109	0.373272233	0.45672364	0.55749767	0.65536627
	STD	0.015962002	0.01438655	0.017270661	0.01745685	0.02266851	0.05633836
Mean value of ALE		0.358741059	0.485956813	0.444548552	0.49942771	0.60244708	0.69891028
Mean value of TOL for 97% CI		±0.063754922	±0.039387391	±0.05234383	±0.02841231	±0.03263037	±0.02891349

Total number of nodes: 100 node. Communication range: 30. Anchor nodes: 20–60%

a significant improvement in accuracy, with an enhancement of 48.69% compared to DV-Hop, 26.22% improvement compared to MGODV-Hop, 19.33% improvement compared to PSODV-Hop, 28.21% improvement compared to WSODV-Hop, and 40.47% improvement compared to SSADV-Hop.

The experiment demonstrates that with fewer than 30 anchor nodes, the average hop distance error increases, and the available localization information decreases, leading to a higher ALE. In contrast, with more than 50 anchor nodes, the accuracy of the average hop distance improves, providing more information for localization and resulting in a lower ALE. However, increasing the number of anchor nodes also incurs additional network costs. Therefore, it is crucial to find a balance between the number of anchor nodes and the acceptable localization error, based on specific application requirements.

In assessing the localization ratio, a threshold-based approach is used to distinguish between successfully and unsuccessfully located nodes. After the localization algorithm is executed, the estimated positions are compared with the actual positions of the nodes. If the error, calculated as the Euclidean distance between the true and estimated positions, exceeds a predefined threshold, the node is categorized as not located; otherwise, it is considered successfully located. In this experiment, successful localization is defined when the error falls within 5% of the communication range, equivalent to 1.5 ms.

The localization ratio of the EMGODV-Hop method is compared with other methods across various anchor ratios in Table 6 and Fig. 7. The results highlight the superior localization capabilities of the EMGODV-Hop method, which achieves a higher localization ratio than its counterparts. By employing 20% anchor nodes, the EMGODV-Hop method showed successful localization with a higher ratio of 0.85, surpassing the ratios achieved by the MGODV-Hop, PSODV-Hop, WSODV-Hop, SSADV-Hop, and DV-Hop methods, which were 0.75, 0.81, 0.76, 0.74, and 0.66, respectively. Furthermore, the EMGODV-Hop method demonstrated efficient localization with 60% anchors, reaching a maximum ratio of 1, while the methods of MGODV-Hop, PSODV-Hop, WSODV-Hop, SSADV-Hop, and DV-Hop yielded lower ratios of 0.91, 0.91, 0.89, 0.87, and 0.7, respectively.

Table 6 Examination of the localization ratio under different ratios of anchors

# anchor nodes	EMGODV-Hop	MGODV-Hop	PSODV-Hop	WSODV-Hop	SSADV-Hop	DV-Hop
20%	0.85	0.75	0.81	0.76	0.74	0.66
30%	0.98	0.79	0.85	0.8	0.77	0.68
40%	1	0.85	0.87	0.84	0.8	0.71
50%	1	0.88	0.88	0.88	0.82	0.74
60%	1	0.91	0.91	0.89	0.87	0.77

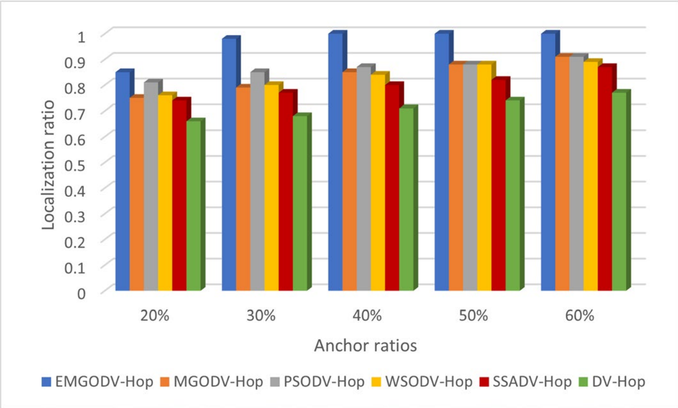


Fig. 7 Investigation of the proposed EMGODV-Hop method's localization ratio under anchor ratio variation

5.2.4 Influence of communication range variation

This experiment aims to evaluate the effect of different communication ranges on the localization accuracy. The experimental configuration consists of 100 unknown nodes and 30 anchor nodes. The communication range varies from 20 m to 60 m, with increments of 10 m. The findings are illustrated in Fig. 8 and summarized in Table 7.

The results illustrated in Fig. 8 demonstrate that increasing the communication range from 20 m to 60 m significantly enhances the accuracy of all localization algorithms. This improvement is due to the greater number of single-hop nodes, which reduces the hop count between nodes and consequently decreases the localization error for unknown nodes. Notably, when the communication range exceeds 50 m, there is no significant effect on ALE for the EMGODV-Hop

Fig. 8 ALE of the comparative algorithm for variety of communication ranges

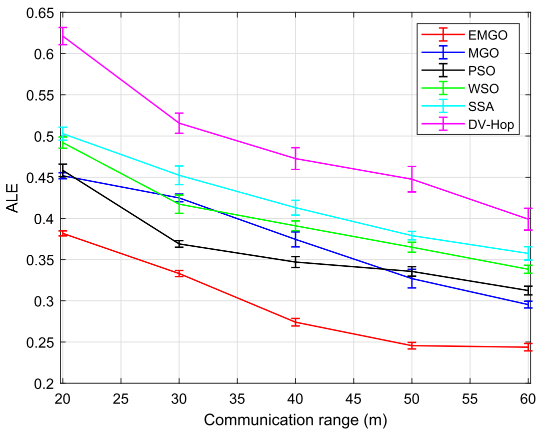


Table 7 Experimental findings of a compared method across a variety of communication ranges

Communication range	Measures	EMGODV-Hop	MGODV-Hop	PSODV-Hop	WSODV-Hop	SSADV-Hop	DV-Hop
20 m	Mean	0.381830068	0.45176269	0.458299512	0.49201026	0.50284847	0.62121435
	STD	0.007725928	0.009466366	0.01936858	0.01710142	0.02003295	0.02671928
30 m	Mean	0.333097429	0.42486044	0.369035718	0.41727035	0.45234695	0.51553769
	STD	0.009321715	0.011862975	0.010083734	0.02793793	0.02839638	0.03079395
40 m	Mean	0.274086849	0.374560682	0.347051098	0.39098065	0.41302611	0.47258445
	STD	0.011755524	0.023227209	0.016989439	0.01526562	0.02333009	0.0338782
50 m	Mean	0.245614969	0.32697971	0.335687317	0.36504624	0.37911662	0.4475635
	STD	0.010114075	0.028870602	0.015167445	0.01556832	0.01312343	0.03969836
60 m	Mean	0.243769942	0.29542844	0.312489834	0.33830342	0.35751516	0.39909079
	STD	0.011515686	0.010511431	0.013425378	0.0124706	0.02078554	0.03394107
Mean value of ALE		0.295685204	0.374753438	0.364417581	0.40063474	0.42096926	0.49119816
Mean value of TOL for 97% CI		±0.052261777	±0.056604608	±0.048857504	±0.05099595	±0.05050047	±0.07292388

Total number of nodes: 100 node. Communication range: 20–60 m. Anchor nodes: 30

algorithm, indicating that its performance remains stable despite further increases in communication range.

Table 7 examines the comparative analysis of EMGODV-Hop in terms of localization error across varying communication ranges. The findings demonstrate that the EMGODV-Hop method achieved the lowest localization error compared to other localization models. Specifically, as the communication range ranged from 20 m to 60 m, the localization error reduced from 0.38, 0.45, 0.45, 0.49, 0.50, and 0.62 to 0.24, 0.29, 0.31, 0.33, 0.35, and 0.40 for EMGODV-Hop, MGODV-Hop, PSODV-Hop, WSODV-Hop, SSADV-Hop, and DV-Hop algorithms, respectively. Moreover, based on the average values of ALE for various localization algorithms and their corresponding tolerance values demonstrated in Table 7, the EMGODV-Hop localization algorithm exhibits superior accuracy, with a 39.8% improvement compared to DV-Hop, 21.09% in comparison to MGODV-Hop, 18.88% in comparison to PSODV-Hop, 26.21% in comparison to WSODV-Hop, and 29.76% in comparison to SSADV-Hop.

5.2.5 Influence of varying the total number of nodes

This section evaluates the performance of the EMGODV-Hop algorithm with varying numbers of nodes and provides a comparative analysis against existing methods. The experiment was conducted with a fixed number of 30 anchor nodes and a communication range of 30 m, with the total number of nodes ranging from 100 to 400. This setup enabled a comprehensive evaluation of the EMGODV-Hop algorithm's effectiveness. The simulation results are presented in Fig. 9 and Table 8.

Figure 9 provides a visual representation of the simulation results, highlighting the accuracy of various algorithms with different numbers of nodes. The results indicate that as the total number of nodes increases, all algorithms show a reduction in localization error. A higher density of unknown nodes improves network connectivity by increasing the average number of nodes within each node's communication range. This results in more direct hop paths between nodes, improving the precision of distance measurements to anchor nodes and reducing localization inaccuracies.

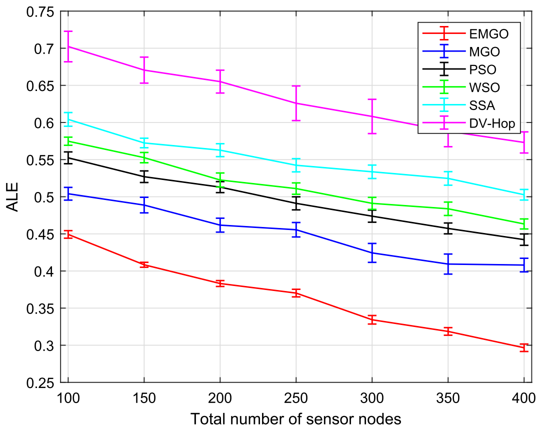


Fig. 9 ALE of the comparative algorithm for different numbers of sensors nodes

Table 8 Comparison of experimental results of the comparative algorithm for different numbers of sensors nodes

# of Nodes	Measures	EMGODV-Hop	MGODV-Hop	PSODV-Hop	WSODV-Hop	SSADV-Hop	DV-Hop
100	Mean	0.449264224	0.504018544	0.552425892	0.57477563	0.60417933	0.70230193
	STD	0.013056536	0.021999765	0.020408122	0.01406583	0.02364566	0.0527505
150	Mean	0.408385248	0.488776244	0.526977831	0.55268714	0.57241389	0.67052881
	STD	0.008375183	0.026498613	0.019680703	0.01746907	0.01622384	0.04421695
200	Mean	0.383135133	0.461742299	0.512971084	0.52253457	0.56272259	0.65505901
	STD	0.010191063	0.024075821	0.019008653	0.02428643	0.02233259	0.03951136
250	Mean	0.370253574	0.455564009	0.491099821	0.51098436	0.54239136	0.62589513
	STD	0.013159292	0.024977482	0.022552146	0.01966548	0.0228757	0.05985117
300	Mean	0.334304805	0.424368547	0.473828894	0.49108016	0.53361155	0.60813527
	STD	0.014737867	0.032686947	0.02083993	0.02071643	0.02320741	0.05921371
350	Mean	0.318618956	0.409332766	0.457329487	0.4837951	0.52467894	0.5882405
	STD	0.01311058	0.034683555	0.018771439	0.02316875	0.02343057	0.0535743
400	Mean	0.296624933	0.407993777	0.442228697	0.46334409	0.50268687	0.57312198
	STD	0.012907195	0.023503887	0.0196798	0.0173457	0.01829967	0.03650585
Mean value of ALE		0.365798125	0.450256598	0.493837387	0.51417158	0.54895493	0.63189752
Mean value of TOL for 97% CI		±0.040543371	±0.028858351	±0.029898359	±0.0298576	±0.02556086	±0.03527449

Total number of nodes: 100–400 node. Communication range: 30. Anchor nodes: 30

However, all six methodologies show a slight decrease in ALE when the total number of nodes increases to 300 or more. This stability can be attributed to the fixed count of anchor nodes, regardless of the overall node count. As more nodes are added, the proportion of anchor nodes decreases, and the information available for locating unknown nodes is insufficient. Consequently, adding excessive nodes may incur additional costs without significantly improving localization accuracy. Therefore, adopting a balanced approach is essential, taking into account the application's specific requirements and constraints to optimize both the localization accuracy and the associated costs.

Table 8 presents a comparative analysis of the EMGODV-Hop model regarding localization error across varying numbers of unknown nodes. The findings highlight the superior performance of the EMGODV-Hop technique compared to other localization models. Specifically, as the node density varied from 100 to 400, the localization error reduced from 0.45, 0.50, 0.55, 0.57, 0.60, and 0.70 to 0.29, 0.40, 0.44, 0.46, 0.50, and 0.57 for EMGODV-Hop, MGODV-Hop, PSODV-Hop, WSODV-Hop, SSADV-Hop, and DV-Hop algorithms, respectively. In addition, based on the average values of ALE for various localization algorithms and their corresponding tolerance values demonstrated in Table 8, the localization accuracy is significantly improved by using the EMGODV-Hop algorithm compared to the DV-Hop, MGODV-Hop, PSODV-Hop, WSODV-Hop, and SSADV-Hop algorithms. On average, the improvement in accuracy achieved by EMGODV-Hop is 40.36%, 16.66%, 23.93%, 26.64%, and 30.90%, respectively.

5.2.6 Computation and communication requirements analysis

This section discusses the computation and communication requirements of the EMGODV-Hop algorithm within the context of WSNs:

The EMGODV-Hop algorithm is designed to be computationally efficient while achieving accurate localization. It includes two main components: an improved DV-Hop algorithm for initial distance estimations and the EMGO algorithm for optimizing node positions.

- Improved DV-Hop phase: This phase focuses on calculating and refining hop distances with minimal computational complexity, utilizing straightforward arithmetic operations that leverage existing node data to adjust and correct distance estimates, minimizing additional computational overhead.
- EMGO optimization phase: This phase employs a metaheuristic approach to fine-tune node positions. The EMGO algorithm is designed for rapid convergence, achieved by strategically balancing exploration and exploitation. This balance is critical because it enables the algorithm to efficiently navigate the solution space to identify optimal or near-optimal solutions without excessive computational effort. Consequently, this efficiency makes the EMGO algorithm particularly effective in resource-constrained environments typical of WSNs.

The EMGODV-Hop algorithm's communication requirements are kept within reasonable limits. Beacon packets are periodically sent by anchor nodes, containing essential information such as node identifiers, coordinates, and hop counts. This periodic transmission is designed to minimize network congestion and communication overhead. Additionally, the algorithm employs a distributed approach where each unknown node performs localization independently based on the information received from nearby anchors. This reduces the overall communication burden compared to centralized methods.

5.3 Discussion

The accuracy of localization algorithms improves as the number of anchor nodes increases, reducing both the distance and hop count between them. This enhancement leads to more precise distance estimations, which are essential for optimizing node localization. This enhancement leads to more precise distance estimations, which are essential for optimizing node localization. The EMGODV-Hop algorithm achieves a lower localization error rate compared to the others. Furthermore, increasing the communication range improves accuracy by decreasing the hop count between nodes. In particular, when the range exceeds 50 m, there is no further effect on ALE for EMGODV-Hop, demonstrating its resilience. Additionally, as the total node count increases, all algorithms exhibit decreased localization error due to improved network connectivity and precision in anchor node distances, resulting in reduced location inaccuracies. This clearly reveals that the proposed EMGODV-Hop method outperforms the other techniques in terms of node localization efficiency.

In any WSN application, the ability to accurately localize nodes sensing specific events is critical. For instance, consider deploying sensor nodes across a farm to monitor environmental conditions and optimize crop yield. Precision in node localization is essential for accurate data collection and analysis in such scenarios. By applying our proposed localization algorithm to agricultural monitoring, farmers can leverage data-driven insights to enhance crop yield, resource conservation, and environmental sustainability. This algorithm also finds application in environmental monitoring contexts like forestry and wildlife conservation, facilitating the monitoring of various environmental factors. In infrastructure monitoring, such as assessing the health of structures like bridges and buildings, our localization algorithm can aid in identifying the exact locations of sensor nodes for precise data collection. Moreover, in industrial automation, our algorithm can support tasks like monitoring industrial processes, tracking assets, and conducting predictive maintenance. These diverse applications underscore the significance and versatility of our node localization algorithm across various domains.

6 Conclusions and future works

WSNs play a vital role in gathering environmental data through sensor nodes. However, accurate localization of these nodes is crucial to interpreting the collected data effectively. Node localization holds significance across various applications, and range-free techniques are favored because of their compatibility with hardware constraints and cost considerations. While the DV-Hop algorithm is a widely known range-free approach for localization, it grapples with accuracy and stability concerns. To address these issues, this paper introduced a novel technique called EMGODV-Hop, designed specifically for localizing nodes in multi-hop networks. This technique comprises two phases: the initial stage utilizes an improved DV-Hop algorithm to estimate distances between unknown and anchor nodes, followed by a second phase, which employs an improved variant of the MGO algorithm, referred to as EMGO, to estimate WSN node positions. The primary aim of this approach is to minimize localization errors and enhance accuracy in determining the coordinates of unknown nodes. The study performed a comprehensive experimental analysis to validate the proposed method's effectiveness and compare the results with other techniques. The findings demonstrated that the proposed EMGODV-Hop method outperforms the other techniques in terms of node localization efficiency. Given the superior performance of our proposed EMGODV-Hop method in terms of node localization efficiency, we believe in its applicability across a diverse range of applications in different domains. In future research, we will focus on enhancing localization performance by investigating advanced algorithms for dynamic network adjustment. Additionally, we will explore integrating factors such as energy efficiency and robustness to varying network conditions into a multi-objective optimization framework. This comprehensive approach aims to improve localization accuracy while simultaneously addressing energy consumption and network resilience.

Data availability Data are available from the authors upon reasonable request.

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Kaur A, Kumar P, Gupta GP (2019) A new localization using single mobile anchor and mesh-based path planning models. Wireless Netw 25:2919–2929
2. Kaur A, Gupta GP, Kumar P (2017) A survey of recent developments in dv-hop localization techniques for wireless sensor network. J Telecommun Electron Comput Eng 9(2):61–71
3. Chaudhary A, Peddoju SK, Kadarla K(2017) Study of internet-of-things messaging protocols used for exchanging data with external sources. In:2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS) pp 666–671. IEEE
4. Aspnes J, Eren T, Goldenberg DK, Morse AS, Whiteley W, Yang YR, Anderson BD, Belhumeur PN (2006) A theory of network localization. IEEE Trans Mob Comput 5(12):1663–1678
5. Kunz Thomas, Tatham Benjamin (2012) Localization in wireless sensor networks and anchor placement. J Sens Actuator Netw 1(1):36–58
6. Ahmad T, Li XJ, Seet BC (2016) 3d localization based on parametric loop division and subdivision surfaces for wireless sensor networks. In:2016 25th Wireless and Optical Communication Conference (WOCC), pp 1–6. IEEE
7. Bachrach J, Taylor C(2005) Localization in sensor networks. *Handbook of sensor networks: Algorithms and Architectures*, pp 277–310

8. Arias Jagoba, Zuloaga Aitzol, Lázaro Jesús, Andreu Jon, Astarloa Armando (2004) Malguki: an rssi based ad hoc location algorithm. Microprocess Microsyst 28(8):403–409
9. Savvides A, Han CC, Strivastava MB (2001) Dynamic fine-grained localization in ad-hoc networks of sensors. In:Proceedings of the 7th Annual International Conference on, Mobile Computing and Networking, pp 166–179
10. Niculescu D, Nath B(2003) Ad hoc positioning system (aps) using aoa. In: IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428), pp 1734–1743. IEEE
11. Niculescu Dragoş, Nath Badri (2003) Dv based positioning in ad hoc networks. Telecommun Syst 22:267–280
12. Bulusu Nirupama, Heidemann John, Estrin Deborah (2000) Gps-less low-cost outdoor localization for very small devices. IEEE Pers Commun 7(5):28–34
13. Nagpal Radhika, Shrobe Howard, Bachrach Jonathan (2003) Organizing a global coordinate system from local information on an ad hoc sensor network. In: Information Processing in Sensor Networks: Second International Workshop, IPSN 2003, Palo Alto, CA, USA, April 22–23, 2003 Proceedings, pp 333–348. Springer
14. He T, Huang C, Blum BM, Stankovic JA, Abdelzaher T(2003) Range-free localization schemes for large scale sensor networks. In: Proceedings of the 9th annual international conference on Mobile computing and networking, pp 81–95
15. Ahmed MM, Houssein EH, Hassanien AE, Taha A, Hassanien E(2018) Maximizing lifetime of wireless sensor networks based on whale optimization algorithm. In: Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017, Springer, pp 724–733
16. Han D, Yu Y, Li KC, de Mello RF (2020) Enhancing the sensor node localization algorithm based on improved dv-hop and de algorithms in wireless sensor networks. Sensors 20(2):343
17. Chen Tianfei, Sun Lijun (2019) A connectivity weighting dv_hop localization algorithm using modified artificial bee colony optimization. J Sens 1–14:2019
18. Cheng BH, Vandenberghe L, Yao K (2010) Distributed algorithm for node localization in wireless ad-hoc networks. ACM Trans Sens Netw 6(1):1–20
19. Annepu Visalakshi, Rajesh Anbazhagan, Bagadi Kalapraveen (2021) Radial basis function-based node localization for unmanned aerial vehicle-assisted 5g wireless sensor networks. Neural Comput Appl 33(19):12333–12346
20. Holland JH (1992) Genetic algorithms. Sci Am 267(1):66–73
21. Kennedy J, Eberhart R(1995) Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, pp 1942–1948
22. Dorigo M, Stützle T(2019) Ant colony optimization: overview and recent advances. Springer
23. Abd El Ghafour MG, Kamel SH, Abouelseoud Y (2021) Improved dv-hop based on squirrel search algorithm for localization in wireless sensor networks. Wireless Netw 27:2743–2759
24. Chai Qing-wei, Chu Shu-Chuan, Pan Jeng-Shyang, Pei Hu, Zheng Wei-min (2020) A parallel woa with two communication strategies applied in dv-hop localization method. EURASIP J Wirel Commun Netw 1–10:2020
25. Zhihua C, Bin S, Yu G, Wang X, Chen J (2017) A novel oriented cuckoo search algorithm to improve dv-hop performance for cyber-physical systems. J Parallel Distrib Comput 103:42–52
26. Chen Xiao, Zhang Benliang (2012) Improved dv-hop node localization algorithm in wireless sensor networks. Int J Distrib Sens Netw 8(8):213980
27. Wang Penghong, Huang Jianrou, Cui Zhihua, Xie Liping, Chen Jinjun (2020) A gaussian error correction multi-objective positioning model with nsga-ii. Concur Comput Pract Exp 32(5):e5464
28. Kanwar Vivek, Kumar Ashok (2021) Range free localization for three dimensional wireless sensor networks using multi objective particle swarm optimization. Wireless Pers Commun 117(2):901–921
29. Wang Penghong, Zhou Rui, Fan Xiaopeng, Zhao Debin (2022) A distance estimation model for dv-hop localization in wsns. IEEE Trans Veh Technol 72(4):5290–5299
30. Wang P, Wang X, Li W, Fan X, Zhao D (2024) Dv-hop localization based on distance estimation using multinode and hop loss in wsns. arXiv preprint arXiv:2403.04365
31. Wang Penghong, Li Hangjuan, Cai Xingjuan (2024) 3d many-objective dv-hop localization model with nsga3. Soft Comput 28(7):6171–6183
32. Abdollahzadeh B, Gharehchopogh FS, Khodadadi N, Mirjalili S (2022) Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Adv Eng Softw 174:103282

33. Osama S, Ali M, Ali AA, Shaban H (2023) Gene selection and tumor identification based on a hybrid of the multi-filter embedded recursive mountain gazelle algorithm. Comput Biol Med 167:107674
34. Izci Davut, Ekinci Serdar, Altalhi Maryam, Daoud Mohammad Sh, Migdady Hazem, Abualigah Laith (2024) A new modified version of mountain gazelle optimization for parameter extraction of photovoltaic models. Electrical Engineering, pp 1–21
35. Alomoush W, Houssein EH, Alrosan A, Abd-Alrazaq A, Alweshah M, Alshinwan M (2024) Joint opposite selection enhanced mountain gazelle optimizer for brain stroke classification. Evol Intel 17:1–19
36. Abd-Elmaboud ME, Saqr AM, El-Rawy M, Al-Arifi N, Ezzeldin R (2024) Evaluation of groundwater potential using ann-based mountain gazelle optimization: a framework to achieve sdgs in east el oweinat, egypt. J Hydrol Reg Stud 52:101703
37. Zellagui M, Belbachir N, El-Sehiemy RA (2023) Solving the optimal power flow problem in power systems using the mountain gazelle algorithm. Eng Proceed 56(1):176
38. Mehrabi Mahshid, Taheri Hassan, Taghdiri Pooria (2017) An improved dv-hop localization algorithm based on evolutionary algorithms. Telecommun Syst 64:639–647
39. Cui Laizhong, Chong Xu, Li Genghui, Ming Zhong, Feng Yuhong, Nan Lu (2018) A high accurate localization algorithm with dv-hop and differential evolution for wireless sensor network. Appl Soft Comput 68:39–52
40. Liu Guiqi, Qian Zhihong, Wang Xue (2019) An improved dv-hop localization algorithm based on hop distances correction. China Commun 16(6):200–214
41. Ling Song, Liqin Zhao, Jin Ye et al (2019) Dv-hop node location algorithm based on gso in wireless sensor networks. J Sens 2019:2986954
42. Li Jianpo, Gao Min, Pan Jeng-Shyang, Chu Shu-Chuan (2021) A parallel compact cat swarm optimization and its application in dv-hop node localization for wireless sensor network. Wireless Netw 27:2081–2101
43. Xianfeng Ou, Meng Wu, Yuanyuan Pu, Bing Tu, Zhang Guoyun, Zhi Xu (2022) Cuckoo search algorithm with fuzzy logic and gauss-cauchy for minimizing localization error of wsn. Appl Soft Comput 125:109211
44. Jia Wenxian, Qi Guohong, Liu Menghan, Zhou Jie (2022) A high accuracy localization algorithm with dv-hop and fruit fly optimization in anisotropic wireless networks. J King Saud Univ Comput Inform Sci 34(10):8102–8111
45. Cao Yuxiao, Jinbao Xu (2023) Dv-hop-based localization algorithm using optimum anchor nodes subsets for wireless sensor network. Ad Hoc Netw 139:103035
46. Sun Haibin, Li Hongxing, Meng Ziran, Wang Dong (2023) An improvement of dv-hop localization algorithm based on improved adaptive genetic algorithm for wireless sensor networks. Wireless Pers Commun 130(3):2149–2173
47. Elie Tagne Fute, Doris-Khöler Nyabeye Pangop, Emmanuel Tonye (2023) A new hybrid localization approach in wireless sensor networks based on particle swarm optimization and tabu search. Appl Intell 53(7):7546–7561
48. Peng Duo, Gao Yuwei (2023) Localization algorithm for anisotropic wireless sensor networks based on the adaptive chaotic slime mold algorithm. Neural Comput Appl 35(36):25291–25306
49. Xue Dalong (2019) Research of localization algorithm for wireless sensor network based on dv-hop. EURASIP J Wirel Commun Netw 1–8:2019
50. Mohamed AW, Hadi AA, Mohamed AK, Awad NH (2020) Evaluating the performance of adaptive gainingsharing knowledge based algorithm on cec 2020 benchmark problems. In: 2020 IEEE Congress on Evolutionary Computation (CEC), pp 1–8. IEEE
51. Hashim FA, Hussain K, Houssein EH, Mabrouk MS, Al-Atabany W (2021) Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Appl Intell 51:1531–1551
52. Faramarzi Afshin, Heidarinejad Mohammad, Stephens Brent, Mirjalili Seyedali (2020) Equilibrium optimizer: a novel optimization algorithm. Knowl Based Syst 191:105190
53. Algorithm and applications (2019) Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi Mafarja, and Huiling Chen. Harris hawks optimization. Futur Gener Comput Syst 97:849–872
54. Zhao Shijie, Zhang Tianran, Ma Shilin, Chen Miao (2022) Dandelion optimizer: a nature-inspired metaheuristic algorithm for engineering applications. Eng Appl Artif Intell 114:105075

55. Mirjalili Seyedali, Lewis Andrew (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
56. Hashim FA, Mostafa RR, Hussien AG, Mirjalili S, Sallam KM (2023) Fick's law algorithm: a physical law-based algorithm for numerical optimization. Knowl Based Syst 260:110146
57. Dehghani Mohammad, Montazeri Zeinab, Trojovská Eva, Trojovskỳ Pavel (2023) Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. Knowl Based Syst 259:110011
58. Derrac Joaquín, García Salvador, Molina Daniel, Herrera Francisco (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18
59. Braik M, Hammouri A, Atwan J, Al-Betar MA, Awadallah MA (2022) White shark optimizer:a novel bio-inspired meta-heuristic algorithm for global optimization problems. Knowl Based Syst 243:108457
60. Seyedali Mirjalili, Gandomi Amir H, Zahra Mirjalili Seyedeh, Shahrzad Saremi, Hossam Faris, Mohammad Mirjalili Seyed (2017) Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191

Authors and Affiliations

Reham R. Mostafa[1,2] · Fatma A. Hashim[3,4] · Ahmed M. Khedr[5] · Zaher AL Aghbari[5] · Imad Afyouni[5] · Ibrahim Kamel[6] · Naveed Ahmed[5]

✉ Reham R. Mostafa
REldeiasti@sharjah.ac.ae; reham_2006@mans.edu.eg

Fatma A. Hashim
fatma_hashim@h-eng.helwan.edu.eg

Ahmed M. Khedr
akhedr@sharjah.ac.ae

Zaher AL Aghbari
zaher@sharjah.ac.ae

Imad Afyouni
iafyouni@sharjah.ac.ae

Ibrahim Kamel
kamel@sharjah.ac.ae

Naveed Ahmed
nahmed@sharjah.ac.ae

1 Big Data Mining and Multimedia Research Group, Centre for Data Analytics and Cybersecurity (CDAC), Research Institute of Sciences and Engineering (RISE), University of Sharjah, Sharjah 27272, United Arab Emirates

2 Department of Information Systems, Faculty of Computers and Information Sciences, Mansoura University, Mansoura 35516, Egypt

3 Faculty of Engineering, Helwan University, Helwan, Egypt

[4] Jadara University Research Center, Jadara University, Irbid, Jordan

[5] Department of Computer Science, University of Sharjah, Sharjah 27272, United Arab Emirates

[6] Department of Computer Engineering, University of Sharjah, Sharjah 27272, United Arab Emirates