Indonesian Journal of Electrical Engineering and Computer Science
Vol. 40, No. 1, October 2025, pp. 461~479
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v40.i1.pp461-479

Optimizing distance vector-hop localization in wireless sensor networks using the grasshopper optimization algorithm

Janani Selvaraj[1], Hymlin Rose Sasijohn Gloryrajabai[2], Sivarathinabala Mariappan[3], Backia Abinaya Antony Samy[4], Sudhakar Kalairishi[1]
[1]Department of Electronics and Communication Engineering, Periyar Maniammai Institute of Science and Technology (Deemed to be University), Vallam, India
[2]Department of Electronics and Communication Engineering, R.M.D Engineering College, Kavaraipettai, India
[3]Department of Electronics and Communication Engineering, Velammal Institute of Technology, Thiruvallur. India
[4]Department of Electronics and Communication Engineering, St. Joseph's College of Engineering and Technology, Thanjavur, India

Article Info

Article history:

Received Jan 10, 2025
Revised Apr 6, 2025
Accepted Jul 4, 2025

Keywords:

DV-hop
Internet of things
Localization
Security
Wireless sensor networks

ABSTRACT

In scenarios involving mobile sensors within distributed sensor systems, such as those often encountered in wireless sensor networks (WSNs) or the internet of things (IoT), the ability to ascertain the origin of sensor data holds significant importance. Range-free Monte Carlo Localization methods offer an energy-efficient solution that eliminates the need for extra hardware, as they solely rely on the radio hardware already present on sensor nodes. But there are certain disadvantages when implemented, as it occupies more amount of power and some inaccuracies might happen in accessing the data from the sensor node. In this paper, we suggest the grasshopper optimization algorithm (GOA) strategy, which incorporates the distance-vector hop (DV-Hop) and three-anchor methods. It displays its usefulness in terms of both overall localization accuracy and resistance to hostile attacks or malfunctioning nodes. Nonetheless, the incorporation of dead reckoning based on motion sensor data significantly enhances the precision of location estimates and bolsters the network's robustness against both faulty components and malicious agents.

Corresponding Author:

Janani Selvaraj
Department of Electronics and Communication Engineering
Periyar Maniammai Institute of Science and Technology (Deemed to be University)
Vallam, Thanjavur 613403, Tamil Nadu, India
Email: jananiassociateprofessor2024@gmail.com

1. INTRODUCTION

A wireless sensor network (WSN) consists of a group of sensors that are strategically placed and designed for the purpose of observing and documenting environmental conditions. These sensors gather information from their respective surroundings and then transmit this data to a central hub for further examination. WSNs possess the ability to measure a wide range of environmental factors, including but not limited to temperature, sound levels, pollution levels, humidity, wind speed, wind direction, and atmospheric pressure, among other variables [1]. Originally designed for military applications, the use of WSNs has expanded to include domains like healthcare, traffic management, and numerous consumer and industrial sectors [2].

A typical WSN consists of tens, hundreds, or even thousands of sensor nodes. A sensor node is typically composed of a minimum of four major components: a radio transceiver with antenna, a microcontroller, electrical interfacing circuits, and a power source that, in most cases, is a battery. Notably, the size of these sensor nodes can vary significantly, ranging from the dimensions of a shoebox to being as small as a grain of dust [3]. Consequently, the cost of these sensor nodes also varies widely, ranging from a few pennies to hundreds of dollars, depending on factors such as energy efficiency, computational speed, bandwidth, and memory capacity, which define the functionality of the sensor [4].

Furthermore, the fact that sensor nodes operate in remote and often challenging environmental conditions makes replacing batteries an impractical solution. Conversely, many applications of sensor networks involve surveillance, necessitating a prolonged operational lifespan. Consequently, a critical research challenge revolves around delivering an energy-efficient surveillance service for specific geographical areas [5], [6]. Current research efforts predominantly concentrate on achieving complete or partial sensing coverage while conserving energy. In this approach, nodes are placed in a dormant state as long as neighboring nodes can maintain sensing coverage on their behalf. These solutions typically treat sensing coverage in a particular geographic area as a binary concept, either providing coverage or not [7].

Nonetheless, we contend that in many scenarios, such as military battlefields, specific geographic zones, like the central command center, hold far greater security significance than others [8], [9]. Acknowledging that individual sensor nodes are susceptible to unreliability and potential failure, and that single sensor readings can be susceptible to interference from background noise, resulting in false alarms, it becomes clear that depending on a single sensor to safeguard a critical area is inadequate. In these situations, a broader coverage approach is necessary, where multiple sensors concurrently monitor the same location to ensure a robust level of confidence in threat detection [10]–[12]. Conversely, delivering the same extensive coverage to less critical areas is not only excessive but also consumes significant energy resources. Middleware operates as an intermediary layer positioned between the operating system and the application. This intermediary layer is critical for enabling communication as well as data transfer between various system components, making it an essential element in optimizing system performance and efficiency.

Middleware developed for WSN must incorporate mechanisms to embed application-specific knowledge into both the infrastructure that supports it and the WSN itself. Data-centric communication necessitates a communication methodology that is similar to content-based messaging systems, rather than typical remote procedure call (RPC)-style communication approaches [13]–[15]. Moreover, event-driven communication is better suited to the fundamental traits of WSN than traditional request-response approaches. In essence, WSN middleware accomplishes a smoother integration of communication and application-specific processing of data compared to traditional systems [16], [17]. The guiding principles of adaptable fidelity algorithms necessitate the infrastructure to offer appropriate mechanisms for determining parameters or even entire algorithms that can efficiently tackle a particular issue while enhancing quality within predefined resource limitations.

This adaptability ensures that the WSN can adjust its behavior and resource utilization based on the evolving needs and conditions of the application, contributing to overall efficiency and performance. The predominant method for addressing this challenge involves employing the global positioning system (GPS). Nevertheless, utilizing GPS comes with a set of drawbacks. GPS sensors tend to be relatively expensive and have high power consumption, which can limit their practicality. Furthermore, GPS sensors are dependent on receiving satellite signals, which renders them ineffective for indoor operations and can lead to reduced precision when used in specific outdoor environments. Addressing the issues related to the expenses and energy consumption of GPS sensors can be tackled through various methods. One proposed solution is to strategically install GPS sensors on a limited number of nodes [18], [19]. These nodes, equipped with GPS sensors, serve as pivotal "seed" or "anchor" nodes, aiding other nodes in determining their positions. An alternative widely used approach is to deploy fixed anchor points at predetermined locations, thereby obviating the necessity for mobile anchoring nodes that incorporate GPS sensors.

A commonly employed method for localization involves utilizing received signal strength (RSS) in combination with an appropriate propagating model to determine the distance that exists between an unidentifiable node and an appropriate reference node. This approach relies on an assumption that RSS decreases corresponding to the distance from the transmitter. In particular, time difference of arrival (TDoA)-based methods require highly accurate clock synchronization among nodes, angle of arrival (AoA)-based methods struggle with issues such as multipath interference and non-line-of-sight (NLoS)scenarios, and they also have to face challenges concerning array calibration. In the case of RSS-based techniques, factors such as radio noise levels, multipath effects, and measurement errors can have an impact on performance [20]. In general, range-based methods often necessitate additional specialized hardware, synchronization of clocks, and increased power consumption to facilitate the active measurements carried out by unknown nodes. Furthermore, the inherent drawbacks associated with all kinds of measurements may affect the preciseness of localization under specific circumstances.

To address these challenges, current research efforts are concentrated on range-free solutions primarily grounded in network connectivity alone [21]. These strategies do not hinge on active measurements conducted by the unknown nodes, rendering them simpler to implement and more cost-efficient [22]. These range-free solutions aim to streamline complexity, reduce hardware requirements, and minimize energy consumption while still achieving reliable localization within wireless sensor networks (WSNs). The proposed algorithm addresses the challenges and issues encountered in existing methods, offering a suitable solution for locating sensor nodes with a small delay and error. WSNs have emerged as a formidable mechanism for the surveillance and acquisition of environmental data across multiple domains, encompassing military applications, healthcare, and industrial sectors. Nonetheless, numerous challenges persist in the optimization of the design, deployment, and management of these networks. These challenges encompass energy efficiency, network scalability, data accuracy, and the reliability of sensor nodes. The constrained power resources of sensor nodes, which are frequently reliant on batteries, necessitate the implementation of energy-efficient algorithms to facilitate prolonged operational longevity. Furthermore, the dynamic and often adverse environmental conditions in which these networks are deployed further exacerbate the performance and reliability of the system. As the quantity of deployed sensor nodes escalates, the assurance of efficient data transmission, the minimization of communication overhead, and the maintenance of low latency become increasingly challenging. Additionally, the integration of heterogeneous sensor types with varying computational capabilities introduces complexity to the system's overall functionality. Consequently, there exists an urgent imperative to address these issues to augment the performance, reliability, and scalability of WSNs within practical applications. This research aims to address the critical challenges faced by WSNs, focusing on enhancing energy efficiency, improving network scalability, and ensuring the reliability and accuracy of environmental data collection. The contributions of this research are as follows:

- Energy-efficient algorithms: development of novel algorithms that optimize energy consumption in WSNs, ensuring longer operational lifespans for sensor nodes without compromising data quality.
- Scalability and network optimization: propose a framework to improve the scalability of WSNs, enabling the efficient management of large sensor networks deployed over expansive areas while minimizing communication overhead.
- Data accuracy and sensor fusion: explore methods for sensor data fusion and error correction techniques to enhance the accuracy of measurements and mitigate the effects of noise and environmental factors on data integrity.
- Robustness in harsh environments: Investigate strategies for improving the robustness of sensor nodes against harsh environmental conditions, ensuring reliable performance even in challenging deployments.

By addressing these issues, the research aims to provide a comprehensive solution to the optimization of WSNs, ensuring that these networks can operate effectively and efficiently across a wide range of applications.

2. RELATED WORKS

The centralized connectivity-based DV-Hop (CCDV-Hop) algorithm [23] with the goal of enhancing the precision of DV-Hop localization. They formulated a problem of optimization that includes the precise connectivity between any two nodes as a limitation. This methodology ensures that the localization results are aligned with the actual network connectivity. Subsequently, they introduced a lower-complexity algorithm known as the distributed connectivity-based DV-Hop (DCDV-Hop) algorithm. DCDV-Hop is capable of providing near-optimal localization efficiency over distributed networks. In contrast to examining the connections of all nodes, the limitations in the DCDV-Hop algorithm specifically concentrate on actual connectivity within a two-hop range.

In a paper authored by his team [24], they present a malicious node detection algorithm (MNDC) in addition to a modified version referred to as EMDC. These algorithms apply density-based spatial clustering to identify unusual clusters within the network. Subsequently, these clusters undergo a sequential probability ratio test to detect undesirable nodes that pose a threat to the network's integrity. The outcomes of the simulation and subsequent analysis demonstrate that the algorithms being suggested surpass other state-of-the-art schemes in terms of their detection accuracy and effectiveness.

In their research, [25] introduced a closed-form solution for source localization. This method combines multi-station time-difference-of-arrival (TDOA) and single-station AOA measurements. A unique feature of this approach is that only the reference station used for TDOA calculation needs to notice the AOA measurements of the origin. This method effectively addresses the challenge of hybrid localization, which involves having various amounts of TDOA and AOA measurements available. Theoretical analysis has confirmed that this approach can achieve performance levels that are very close to the Cramer-Rao lower bound (CRLB) when noise levels are minimal. Furthermore, the study models the connection between the positions of each station in regard to the source and the accuracy of localization. By applying the Doptimality criterion, this research determines the optimal geometric arrangement among multiple stations and the source.

An existing paper provides an algorithm called Sequential DV-Hop that is based on the multihop wireless network of sensors. This improves the localization performance step by step by calculating the node locations. Variability in the number of accessible anchors within the network is taken into account by the method. To formulate the DV-Hop algorithm as a node localization basis, a new and efficient technique for computing the average distance between nodes' hops was used. Gui *el al.* [26] proposed an algorithm of DV-Hop based on the cyclotomic method and weighted normalization called CMWN-DV-Hop, where segmentation and weighting factors are introduced and then normalized. For calculating the coordinates of unknown nodes, they adapted the weighted Recursive Least-Squares algorithm, referred to as WRLS. Based on the aforementioned requirements, they performed comprehensive testing to benchmark the algorithm's performance with respect to varying conditions such as different numbers of nodes, varying ratios of anchor nodes, and varying communication radii. Simulations have proven that localization error is significantly mitigated by the proposed algorithm.

Gupta and Singh [27] proved an enhanced DV-Hop localization algorithm which uses a number of communication radii. It calculates through the mean hop distance with the use of the cosine theorem while correcting the hopping count through maximum predicted distance of unknown nodes. The method applies different communication radii for broadcasting positions to minimize the number of hops required between beacon nodes and unknown nodes. There, it uses either the maximum likelihood estimates or the trilateration method to find the coordinates of the location after estimating the appropriate hop distance with the help of the cosine theorem.

Hosseinzadeh *et al.* [28] proposed a unique DV-Hop location algorithm centred around the concept of half-measure weighted centroid. This novel approach takes into account the two-dimensional in nature position distribution and begins by constructing the minimal communication radius, resulting in a logical network interconnection framework. The method then fine-tunes the distance between beacon nodes and their nearby nodes to improve jump distances, eventually optimizing the shortest path for more precise localization.

The proposed localization algorithm is then theoretically analyzed and validated with simulated trials. These studies investigate a variety of circumstances, including the usage of the precise same communication radius, multiple communication radii, and varied densities of nodes within the same communication radius. To evaluate the performance of the newly suggested method, the study compares it to the usual DV-Hop localization technique in terms of localization error and accuracy. While the innovative strategy that employs the grasshopper optimization algorithm (GOA) to elevate distance vector-hop localization (DV-HL) in wireless sensor networks (WSNs) showcases exciting prospects in terms of precision, energy conservation, and adaptability, there are several challenges to ponder. Firstly, even though GOA has proven its prowess in medium-sized WSNs, its ability to scale up to colossal networks with countless nodes poses a significant hurdle, as the computational intricacies may escalate and hinder real-time performance. Furthermore, the fidelity of DV-HL localization is heavily reliant on the initial configuration of anchor nodes, and in practical contexts, achieving the ideal anchor arrangement can be quite challenging, potentially compromising efficiency. The research also presumes perfect conditions for communication and distance assessment; however, real-world elements like signal disruption and background noise could result in increased localization inaccuracies. Although GOA aims to optimize anchor node positioning to diminish energy usage, vast networks may still grapple with energy inefficiencies owing to the algorithm's iterative nature. In addition, the real-time execution of the algorithm might encounter latency challenges, particularly in time-critical applications such as industrial IoT systems, where delays are simply intolerable. The findings of the study are derived from simulations conducted under optimal circumstances, and actual field trials would be essential to further authenticate the method, taking into account the varied environmental influences and sensor node traits. Lastly, the presumption of uniform sensor nodes with comparable capabilities overlooks potential discrepancies in node specifications, which could affect the algorithm's efficacy in diverse networks. These constraints illuminate pathways for future exploration, wherein additional refinement and customization can amplify the approach’s real-world relevance.

These research papers serve as motivation for our proposed paper, where we aim to introduce a localization approach based on the GOA in conjunction with the DV-Hop algorithm. Our objective is to compute the locations of randomly deployed target nodes and evaluate the performance of this approach based on a set of predefined parameters. While genetic algorithms and particle swarm optimization are powerful metaheuristics with broad applications, Grey Wolf Optimization was chosen for this research due to its strong balance of exploration and exploitation, fewer tuning parameters, robustness in convergence, and efficiency in high-dimensional optimization tasks. These advantages make GOA particularly well-suited to the optimization challenges encountered in WSNs and similar complex real-world applications. In problems involving large datasets or high-dimensional search spaces, GOA tends to outperform both GA and PSO in terms of efficiency. The dynamic nature of the grey wolf pack's leadership hierarchy allows for better exploration of high-dimensional spaces and reduces the likelihood of getting trapped in regions that do not contribute to optimal solutions. Recent studies have made notable advancements in optimizing localization and routing protocols within WSNs. While this research focuses on the integration of the GOA with DV-HL, it is crucial to compare these findings with the latest works to highlight the novelty and contribution of this study. By comparing with these recent studies, we find that the GOA offers significant improvements over traditional algorithms in both localization accuracy and energy efficiency in WSNs. While hybrid and multi-algorithmic approaches have been successful in specific contexts, GOA proves to be a more efficient, simpler alternative that still achieves competitive performance. Moreover, the application of GOA to DV-HL localization presents a novel approach that fills the gap in current research by optimizing both network configuration and node placement in large-scale environments.

3. LOCALIZATION BASED ON DV-HOP ALGORITHM

One popular range-free localization method in WSN is the distance-vector hop (DV-Hop) algorithm. By calculating the number of hops needed to get from one node to the other, this technique determines the distance between two nodes. The shortest path first is the method that DV-Hop uses to function. Individual nodes in WSNs using DV-Hop determine how far apart they are from other nodes in the network, then broadcast this distance information to other nodes [23], [24].

This information exchange enables beacon nodes, which typically have known positions, to receive data about the distances to other nodes and subsequently determine the minimum hop counts required to reach those positions. This way, DV-Hop aids in the localization of nodes within the network, even without the need for precise distance-measuring hardware like GPS receivers. The DV-Hop algorithm, thus accounts for the change in the anchor nodes of the network. In this process, every one of its anchor nodes sends a data packet throughout the network so that upon reception by other anchor nodes, they increment, in the hop count field by one unit. With that information, an anchor node then calculates the average number of hops which it is away from other anchor nodes of the network.

$$Hopsize_i = \frac{\sum_{j\neq i}\sqrt{(p_i-p_j)^2+(q_i-q_j)^2}}{\sum_{j\neq i}h_{ij}} \tag{1}$$

In the wider context of the DV-Hop algorithm, where (p_i, q_i) and (p_j, q_j) indicate the precise locations of the two arbitrary anchor nodes, h_{ij} is the total quantity of hops between these anchor nodes, and Hopsizei is a mean number of hops between the anchor nodes, for every sensor node in the overall network computes the average number of hops to anchor nodes. The predicted distance is calculated from the total amount of hops (h_{ij}) to the anchor node and the average number of hops (Hopsizei) in between the anchor nodes. This algorithm enables sensor nodes to determine their distance from anchor nodes, which helps with network localization.

$$hoppk * hopsizei \tag{2}$$

Figure 1 illustrates the process of sending and receiving hops to estimate distances in the DV-Hop algorithm. While DV-Hop offers simplicity, cost-effectiveness (as it doesn't require range information), and reduced cognitive complexity, it is known for its insufficient reliability, especially in low-density systems. In the realm of the internet of things (IoT), the following mechanism is employed to ascertain distances between the objects, whether they are equipped with GPS or not:

- Objects with GPS: for objects that are equipped with GPS, they compute the distance, between one another and subsequently establish the number of hops required to connect with one another.
- Computation of average hops and distances: within the IoT environment, an average amount of hops and distances are computed. Objects lacking GPS capability then calculate the number of hops required based on the information obtained from objects with known locations, thereby enabling their own localization.
- Distance estimation: the number of hops multiplied by the average distance can be used to estimate the distances between objects, whether or not they have GPS.

While this approach is more accurate, it increases overhead for networks because it requires sending a large number of packets at various network stages. The correctness of this approach hinges on the density and uniform distribution of network nodes. Greater node density and a more even distribution typically enhance the accuracy of this distance estimation technique. In the discussion section, it is imperative to examine the trade-offs inherent in the enhancement of accuracy juxtaposed with the escalation of computational complexity. Although the proposed GOA markedly enhances localization precision within

WSNs, it concomitantly imposes a greater computational burden attributable to its iterative optimization methodology. This augmented complexity may pose significant challenges in real-time applications, particularly within expansive IoT networks where computational resources are constrained. The algorithm's capacity to sustain performance while concurrently ensuring that the computational expense remains within feasible limits is a critical factor for its pragmatic deployment. In this regard, further empirical investigation is essential to ascertain the computational thresholds of GOA and to evaluate its efficacy in real-time operational contexts.

Moreover, the algorithm's capacity for adaptability in response to dynamic or mobile sensor nodes warrants thorough examination. In the domains of IoT and IIoT applications, sensor nodes are frequently subject to mobility, engendering complexities in both localization accuracy and energy management. It would be advantageous to investigate the potential for extending the GOA-DV-HL framework to accommodate the challenges posed by dynamic node movement. The algorithm may be amenable to modifications that optimize trajectory planning for mobile sensor nodes or mobile sinks, thereby ensuring the preservation of localization precision as nodes transit through the network. Addressing these dynamic circumstances will significantly enhance the robustness and applicability of the algorithm in practical, real-world situations.

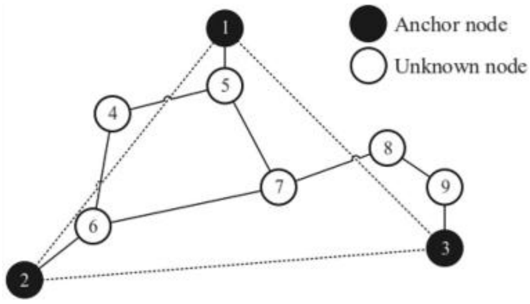

Figure 1. DV-Hop algorithm

3.1. Three-anchor method

The Figure 2 shows a three-anchor location technique that uses the coordinates of the three anchor nodes in order to determine the locations of smart objects. More hops typically indicate longer distances between nodes in the context of networked computers. In the context of the IoT, a smart node without GPS can calculate its distance from other nodes by measuring the number of hops required to connect.

Within themethod of three-anchor, a smart node without GPS functionality computes the hop count needed to reach nodes equipped with GPS within the IoT network. Leveraging this hop count information, it can approximate its distance from the GPS-enabled nodes and subsequently determine its own location using the three-anchor method. To effectively implement this approach, it is imperative to possess the coordinates of at least three intelligent elements within the network. The problem occurs when these three reference sites might not be located within the intelligent elements or the item of interest's radio range. Triangulation and other distance estimate methods are therefore required.

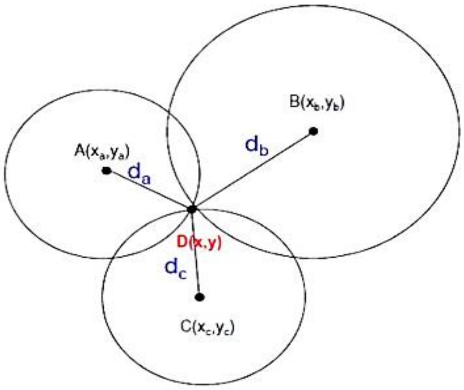

Figure 2. Localization of three anchors

Let's assume that Si, with unknown coordinates (p, q), represents a smart object without GPS. There are at least three GPS-enabled smart objects in the area of Si, identified by the notation A = {A1, A2, A3}, and their coordinates are {(p_1, q_1), (p_2, q_2), (p_3, q_3)}. The formula below can be used to determine the distances from Si to these GPS-equipped objects, {d1, d2, d3}:

$$\sqrt{(p - q_1)^2 + (p - q_1)^2} = d_1 \tag{3}$$

$$\sqrt{(p - q_2)^2 + (p - q)^2} = d_2 \tag{4}$$

$$\sqrt{(p - q_3)^2 + (p - q_3)^2} = d_3 \tag{5}$$

By measuring these distances, Si can estimate its own location (x, y) according to the positions of the GPS-equipped objects and the three-anchor method. These three equations can be squared and redefined as the equation above. The squared term can then be removed by expanding them. This is accomplished by simplifying the first and second equations by subtracting the third equation, leaving two equations.

$$(p - p_1)^2 + (q - q)^2 = {d_1}^2 \tag{6}$$

$$(p - p_2)^2 + (q - q)^2 = {d_2}^2 \tag{7}$$

$$(p - p_3)^2 + (q - q_3)^2 = {d_3}^2 \tag{8}$$

$$(p_1 - p)^2 - (p_3 - p)^2 + (q - q)^2 - (q_3 - q)^2 = {d_1}^2 - {d_3}^2 \tag{9}$$

$$(p_2 - p)^2 - (p_3 - p)^2 + (q_2 - q)^2 - (q_3 - q)^2 = {d_2}^2 - {d_3}^2 \tag{10}$$

The exponential factors y and x in this equation can be removed to produce an equation and a system of linear equations that are simple to solve

$$(p_3 - p_1).p + (q_3 - q_1).q = \frac{({d_1}^2 - {d_3}^2) - ({p_1}^2 - {p_3}^2) - ({q_1}^2 - {q_3}^2)}{2} \tag{11}$$

$$(p_3 - p_2).p + (q_3 - q_2).q = \frac{({d_2}^2 - {d_3}^2) - ({p_2}^2 - p^2) - ({q_2}^2 - {q_3}^2)}{2} \tag{12}$$

indeed, the new undetermined position (p, q) of a smart object can be determined using the given equations. By solving these equations, the smart object can determine its precise location within the IoT network. This process is a fundamental concept in localization techniques, especially in scenarios where GPS is not available or practical, and it allows smart objects to determine their positions based on the distances to known anchor nodes.

3.2. Grasshopper optimization algorithm (GOA)

The GOA is based on grasshopper populations' migratory habits in search of food-rich sources, representation of GOA is shown in Figure 3. This algorithm is designed to mimic the collective and social behaviors seen in mature grasshoppers, which are most pronounced during their mature stage. Grasshoppers are directed by a variety of elements when in flight and on their way to food sources, including wind, gravity, and a natural desire to fly to areas where other grasshoppers congregate. The equation representing the measure of a grasshopper in the GOA algorithm, taking into account the influence of wind, gravity, and the search for optimal positions, is as follows:

$$Xi = Si + Gi + Ai \tag{13}$$

In this equation: represents the point vector for the grasshopper (or a solution representing an optimal point), Si corresponds to the change in position due to the optimal social position vector, Gi represents the effect of gravity on the grasshopper's movement, Ai denotes the influence of wind direction.

This equation illustrates how a grasshopper's position or a solution's state is updated based on these three factors: optimal social position, gravity, and wind direction. The GOA each solution as if it were a coded grasshopper, with the ability to navigate through the problem domain in search of the optimal food source location using the target position.

The vectors Si, Gi, and Ai, which represent the optimum societal position vector, gravity vector, wind vector, can be mathematically described using the following equations, serving as the foundation for the GOA:

$$S_i = \sum_{j=1}^{N} s(d_{ij}).\widehat{d_{ij}} \tag{14}$$

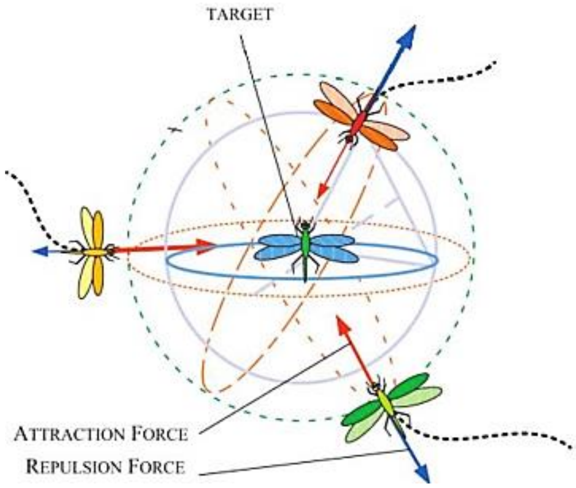

Figure 3. Representation of GOA

$$G_i = -g\widehat{e_g} \tag{15}$$

$$A_i = +u\widehat{e_w} \tag{16}$$

let, "d_{ij}" represents the distance that exists between two arbitrary grasshoppers."dc_{ij}" denotes the unit vector used for a grasshopper's movement,"$s(d_{ij})$" is the jump control function employed by a grasshopper.

$$d_{ij} = |X_i - X_j| \tag{17}$$

$$\widehat{d_{ij}} = \frac{x_i - x_j}{d_{ij}} \tag{18}$$

$$S(d_{ij}) = fe^{\frac{-r}{l}} - e^{-r} \tag{19}$$

The provided information explains the mechanics of the GOA and the various forces and factors that influence grasshoppers' movement within the problem domain:

- Jump controller function: the equation mentions the jump controller function s(dij), which controls the distance between two grasshoppers (Xi and Xj) at their ith and jth positions. This function regulates the jump behavior of grasshoppers in the GOA.
- PC value: the PC value represents the current operation performed and is encumbered with a set point based on jumping directives. It receives additional instructions after executing a specific command.
- Coefficients of attraction (f) and magnitude (l): the coefficients f and l are linked with a grasshopper's attraction to an ideal solution, as well as the size or scale of a grasshopper's jump. These parameters determine how strongly a grasshopper is attracted to an optimal point and how far it can jump in the solution space.
- Distance (r): the variable r represents the separation distance between two grasshoppers (i and j) and represents plays a role in calculating the forces influencing their movement.
- Attraction and repulsion forces: grasshoppers exhibit two primary types of reactions when navigating the problem domain: attraction forces and repulsion forces. Attraction forces pull grasshoppers toward optimal solutions (the target), as indicated by the red vector. Repulsion forces, shown by the blue arrow, cause grasshoppers to explore the region surrounding the target. Some grasshoppers conduct a local search close to the target, while others explore the global space farther from the target.

- GOA evaluation results: according to the evaluation results, the GOA demonstrates superior performance in terms of target accuracy compared to other optimization algorithms such as genetic algorithms, Bat, firefly, and the optimization of particle swarm algorithms, as well as gravitational search.

The GOA leverages these forces and behaviors inspired by grasshopper populations to optimize solutions within the problem domain, making it a promising approach for various optimization tasks.

3.3. Localization method

Localization is a critical aspect of various applications because knowing the precise positions and locations of objects and sensors adds significant value to data and information. In scenarios like emergencies, disasters, or search and rescue operations, having accurate localization information can be crucial. Furthermore, in the context of the IoT, localization is essential for routing algorithms and efficient data management. Developing low-cost localization algorithms with minimal error rates is a key research area.

In this study, a novel and cost-effective localization technique is proposed to accurately identify the positions of network devices. This approach utilizes tags attached to objects to determine their positions and locations. Here's how it works:

- Tag-based localization: localization is achieved through the use of object tags, which actively seek out three anchor objects equipped with GPS in their proximity.
- Distance calculation: after locating these three anchor objects, the tags proceed to compute their respective distances from them by leveraging GPS data.
- Position determination: utilizing the GPS-derived positions of the three anchor objects, the tags then establish their own locations and positions within the network.

However, there is an inherent problem with this approach: there may be situations in which fewer than three GPS-equipped items are within a tag's radio wave range. To tackle this issue, the research utilizes hop-related data. When there are fewer than three smart things equipped with GPS near a site of interest, the system can calculate the distance between them and those smart objects by counting the number of hops. This hop-dependent estimation of distance is accomplished by employing distance regression methods that rely on the hop count data acquired from the DV-Hop algorithm.

The described approach indeed enhances the robustness of localization, especially in situations where GPS-equipped anchor objects are not readily accessible. By leveraging hop information and distance regression techniques, the goal of the study is to improve the precision as well as reliability of the localization within IoT networks. However, it's important to acknowledge that hop-based algorithms like DV-Hop have limitations, primarily related to precision and power usage:

- Precision limitations: errors can occur when converting hop counts into distance units, which can affect the precision of the localization. This is a known challenge with DV-Hop and similar hop-based algorithms.
- High-power usage: each anchor node in DV-Hop must determine the network hop size and communicate that estimate to other nodes. This leads to preventable high-power usage, which can be problematic, particularly in IoT applications with limited battery life.

Regression analysis is used to determine the relationship between variables, with the goal of creating a model that can predict experimental outcomes based on learning data. This analysis identifies the best-fitting line or curve that represents the relationship between variables, enabling the system to make predictions. The proposed method uses the DV-Hop algorithm for localization, which is highly reactive to the network density and may perform less accurately in situations with different node densities. Additionally, when localization is accomplished through distance regression and hop counts, similar challenges related to accuracy may arise, akin to those encountered in the DV-Hop algorithm. Therefore, while the approach aims to improve localization in IoT networks, it's important to address these precision and power usage concerns to ensure the reliability and effectiveness of the localization technique.

Consequently, there could be a decline in localization accuracy. In order to solve the localization error, which is regarded as an issue of optimization, metaheuristic techniques are advised. In an optimization problem, the aim is to find the most effective solution, typically the one that minimizes the objective function within the range that is feasible. Given the exceptionally high accuracy of swarm intelligence systems across various metaheuristic methods, the proposed approach utilizes metaheuristic techniques to mitigate localization errors. When it comes to IoT-based localization, a significant challenge arises from the impracticality of installing GPS sensors on all objects due to the associated implementation costs. Additionally, GPS signals often fail to function correctly indoors, such as in stores and warehouses, leading to increased localization errors.

Error localization, in this context, refers to the process of determining which fields should be updated for a record that has rejected an update. Typically, an optimization method is employed to determine the minimal set of fields that need modification to ensure that the updated record is accepted.

One cost-effective localization method involves leveraging minor object features to determine their positions. For example, it is feasible to affix low-priced tags like radio frequency identification (RFID) tags to all objects. These tags can contain information about the objects, including product prices, and can be used for stationary objects by indicating each object's position. However, in most IoT applications, smart objects can move within a defined area, such as a store or warehouse. The proposed method operates on the assumption that each object is equipped with an RFID tag that can be activated by localization waves and follows these procedures for localization:
– Initialization: radio waves are used to prompt a non-GPS-enabled object to determine its present location and its position coordinates, maybe utilizing IoT infrastructure when required.
– RFID activation: when an object's RFID tag is active, it sends signals to other surrounding GPS-equipped objects. 3Reaction from GPS-Enabled items: When nearby smart objects have GPS, they react to the radio waves by sending back response signals to the initial object.
– In situations where an object lacks knowledge of its coordinates, it computes its travel distance from three GPS-equipped smart objects by means of signal transmission and reception. 5. To determine its distances from at least three GPS-equipped objects, the item of interest uses the three-anchor approach to compute its position and location.

3.4. Steps involved in the proposed system

The suggested strategy presents a novel localization technique for objects connected to the IoT that combines mathematical and geometric equations with signal hop-based distance estimation. To enhance localization accuracy, the GOA is employed. When there are at least three smart objects with GPS capability in proximity to a smart object lacking GPS, it can accurately determine its position by calculating distances from these three nodes based on signal transmission and reception times. It can determine its own position by knowing the locations of a minimum three smart objects and their distances from one another.

However, in circumstances when there are less than three smart objects having GPS capabilities in proximity to the smart objects lacking GPS, the objects are unable to use the same localization mechanism. In such cases, it is critical to estimate the travel distance away from smart objects equipped with GPS and tally the quantity of signal hops required. Hop count is a critical component of distance estimation, much like the DV-Hop algorithm. The IoT localization process includes the following steps:
– Initialization: using tag readers to start the localization process for a smart object that doesn't have position or location data.
– Local survey: the smart object conducts a survey of its immediate vicinity and calculates its distance from nearby smart objects, typically one or two objects in close proximity.
– Beyond radio range: to measure the distance away from other smart objects with GPS capabilities located beyond its radio range, the smart object employs a method similar to the DV-Hop algorithm, involving the transmission of hops and subsequent distance estimation.
– Flooding packets: to calculate its distance from other smart objects, the object of interest sends out a series of flooding packets across the network. The calculation is dependent on the number of hops as well as the presence of GPS-enabled smart devices.
– Geometric localization: the geometric three-anchor approach is used to establish an object's position and location through determining its distance from three smart objects with GPS and known positions.

It's crucial to highlight that the calculated location of the smart object may contain notable inaccuracies, which can be alleviated through the application of metaheuristic techniques. In this specific study, the choice fell on the GOA due to its superior accuracy when compared to other widely recognized algorithms such as genetic algorithms, particle swarm optimization, bat algorithms, firefly algorithms, and gravitational search algorithms. The GOA is firmly grounded in Newton's law of planetary motion, drawing inspiration from real-world processes. In the past decade, researchers have developed various adaptations of the centrifugal search method, fine-tuning its parameters to effectively address intricate objective functions. The GOA generates fresh positions in close proximity to those obtained through the three-anchor method, thus presenting new possibilities for smart object localization with reduced error.

3.5. Assessing the distance between a non-GPS entity and GPS entities using the suggested approach

In the proposed approach, when a non-GPS-equipped object demands localization, it relies on obtaining information. In the context of localizing objects, specifically those that incorporate GPS technology, the process involves determining the precise locations of three GPS-equipped objects and their respective distances. However, there are scenarios where obtaining these three distances directly is not feasible due to the absence of GPS-equipped objects within the required range. In such instances, the hop and distance regression technique can be utilized. Here's a breakdown of how this method operates:

- Activating an RFID tag on a non-GPS object detects nearby GPS-equipped devices, and the system leverages signal exchange with three such GPS-equipped devices to estimate their distance if they are in range. However, the hop and distance regression approach are used if three GPS-equipped items cannot be located in close proximity.
- A GPS-equipped item initiates the transmission of packets that are multicast throughout the network, therefore identifying the number of hops between itself and other nodes and objects.
- The GPS-equipped object subsequently computes its distance from other GPS-equipped objects, using the number of hops as a reference for calculating distance, with an enhanced level of accuracy achieved through average distance calculations.
- Objects that lack GPS capabilities but are "smart" determine the number of hops between GPS-equipped objects.
- Calculating the number of hops between non-GPS and GPS-equipped items makes it easier to estimate their distance.
- The three-anchor approach can be used to determine the location and position of a non-GPS object by calculating the distance that exists between it and GPS-equipped devices.

This approach empowers non-GPS objects to ascertain their positions, even when obtaining the required distances from GPS-equipped objects directly is not possible. This enhances the robustness and versatility of the localization process.

3.5.1. Three-anchor localization

In the proposed approach, a smart object begins the localization process by trying to find three anchor nodes into its radio range. Once these anchor nodes are located, the smart object sends signals through each of them and subsequently receives feedback that is delayed signals. By analysing the time delay of these signals, the smart object can calculate the length of distance from these three anchor nodes. Knowing the positions of these anchor nodes enables the smart object to compute and calculate its own position and location.

In the context of this localization challenge, let us consider that the smart object's position in the IoT can be expressed as (p, q), which are the unknown coordinates that must be found. The smart object in question is characterized by its knowledge of the placements of the three anchor nodes (p_1, q_1), (p_2, q_2), (p_3, q_3). It may calculate its distances from these nodes as d1, d2, and d3, respectively.

In order to solve the linear equation system, the sensor node that is located at position (p, q) in a wireless sensor network (WSN) determines its Euclidean measurement distance from the three anchor nodes. This is achieved by applying the principles of a linear equation represented as Y = mx + b, where 'b' denotes the y-intercept of the graphed line, and 'm' represents the line's gradient. The gradient of a line essentially signifies its steepness or slope. In this context, the gradient signifies the slope of the wall.

Euclidean distance is a fundamental concept in arithmetic, signifying the straight-line distance between two points. In Cartesian coordinates, the Euclidean distance is determined as the length of the line joining the two corresponding points. It is the absolute separation in Euclidean space between two points, such as A and B.

To compute the Euclidean distance between two locations, such as point (p, q) and an anchor node, you can use mathematical functions like the numpy method "linalg.norm." In (19) is applied to calculate the coordinates of the smart object (p, q) in the IoT, leveraging the known distances to anchor nodes and their positions as critical components in the localization process

$$\begin{bmatrix} p \\ q \end{bmatrix} = \frac{1}{2} . \begin{bmatrix} \left(d_1{}^2 - d_3{}^2\right) - \left(p_1{}^2 - p^2\right) - \left(q^2 - q_3{}^2\right) \\ \left(d_2{}^2 - d_3{}^2\right) - \left(p_2{}^2 - p_3{}^2\right) - \left(q_2{}^2 - q_3{}^2\right) \end{bmatrix} . \begin{bmatrix} p_3 - p_1 & q_3 - q_1 \\ p_3 - p_2 & q - q_2 \end{bmatrix}^{-1} \tag{20}$$

this equation uses anchor node positions and observed distances inside the IoT to calculate the position of an unnamed smart object.

3.5.2. Minimizing error in three-anchor localization

The three-anchor approach can be combined with the distance and hop measurement methodology to provide a partial compute of the position and location coordinates of the smart device within the IoT. However, it is critical to understand that this method may be prone to considerable inaccuracies, owing to the intrinsic constraints of the DV-Hop model, which is not known for its high accuracy. In fact, the network's node density has a significant impact on the DV-Hop model's accuracy. The following equation can be used to determine a set of coordinates within a specified radius (R) surrounding a smart object if its current estimated location within the IoT is represented using the three-anchor approach as (x, y):

$$\begin{bmatrix} p' \\ q' \end{bmatrix} = \begin{bmatrix} p + (p \pm R \times rand(0,1) \\ q + (q \pm R) \times rand(0,1) \end{bmatrix} \tag{21}$$

the set {(p_1, q_1), (p_2, q_2), (p_N, q_N)} represents the new coordinates produced near (p, q) that are utilized as a member of the population in the GOA. In (22) can be used to evaluate each of these solutions:

$$J = \sqrt{\Delta x^2 + \Delta y^2} - v.\Delta t \tag{22}$$

The distance that exists between a member of the GOA and a GPS anchor node is represented by p=Δx2 + Δy2, whereas v.Δt represents the distance that lies between a solution or position of the anchor node based on the time it takes for a signal to depart the smart object, meet the anchor node, and return. Prior to implementing the GOA, the proposed method considers a population of solutions. This population of solutions forms the foundation for the GOA to search for an optimal solution, consequently diminishing localization errors. Within this framework, there exists a predefined range referred to as the "agreeable range," denoted as "d = 2.079." This range signifies the distances between two grasshoppers at which they either experience attraction or repulsion. More precisely, when the distance between two grasshoppers is less than "d = 2.079," they repel each other, whereas interaction takes place when the distance exceeds "d = 2.079."

To initiate the GOA, a set of coordinates is produced in 2D space, resulting in a population of grasshoppers. Each grasshopper in this population is then subjected to the localization function that is objective for evaluation to determine optimal positions. In the proposed approach, the GOA is employed to improve and fine-tune the localization of a smart item without GPS capability. The GPS is a satellite-based system for navigation that synchronizes time, speed, and location information for global usage. It is widely used and integrated into various items such as watches, cell phones, and automobiles. A GPS receiver estimates its own position by measuring the time it takes for signals from at least four satellites to reach it.

In the IoT environment, both GPS-equipped and non-GPS-equipped smart objects are initially distributed. Smart objects with GPS capabilities use multicasting to determine their hop count from other GPS-enabled things and send their whereabouts to all other objects. The average hop count is then determined using the DV-Hop algorithm. Non-GPS-equipped devices calculate hop counts from the minimum of three GPS-equipped objects by transmitting packets across the network. Objects having GPS capabilities use their hop counts from other GPS-equipped objects, the average hop count, and distance information to calculate their distance from those objects.

Non-GPS-equipped objects rely on their calculated distances from three GPS-equipped smart objects and the known positions of these GPS-equipped objects to estimate their own positions. However, this technique is prone to error. As a result, a few alternate positions are generated close to the projected position. These alternative positions are used as the initial population for the GOA, facilitating the optimization process to improve the accuracy of smart object localization.

$$Pop = \{(p_1, q_1), (p_2, q_2), \dots\dots (p_N, q_N)\} \tag{23}$$

where (p_i, q_i) is a GOA population member analysed by the evaluation function J for fitness. This position is utilized to update and eliminate localization mistakes using the following equation

$$J(p_i, q_i) = c.\left[\sum_{j=1}^{N} c.\frac{ub_d - lb_d}{2} s\left(\left|(p_i, q_i) - (p_j, q_j)\right|\right).\frac{(p_i, q) - (p_j, q_j)}{d_j}\right] + (p^*, q^*) \tag{24}$$

In this equation, (p∗, q∗) denotes the ideal grasshopper's position or coordinates within the population. Furthermore, ubd and lbd are the upper and lower boundaries of the objective function's dimension 'd', respectively, and 'c' is the rate of convergence coefficient utilized in the Global Optimization Algorithm (GOA). In mathematics, convergence is a property of some sequences and processes in which values approach a maximum or minimum as a variable within a function changes or the number of variables in a sequence grows. The following equation defines convergence as a decreasing function of iteration:

$$c = c_{max} - l\frac{c_{max} - c_{min}}{L} \tag{25}$$

In the provided equation, 'l' denotes the GOAiteration that is currently in progress, and 'L' is the maximum iteration limit. Additionally, the values of 'c_max' and 'c_min' are set to one and 0.0001, respectively. In order to minimize location error, the GOA is carried out iteratively, and with each iteration, the average localization error falls. The average localization accuracy for individual objects is around 0.43

meters when the wireless communication distance is 50 meters, the RSS Indicator (RSSI) variance is 8 dBm, and the percentage of base stations is 1.25%. It's important to note that globalization can be influenced by various factors, including climatic, economic, educational, historical, sociological, geopolitical, legal, and technological factors, among others. In the final iteration of the GOA, the position of the ideal grasshopper is chosen as the definitive and optimal position for the non-GPS-equipped object, boosting localization accuracy.

4. SIMULATION RESULT

The two specific examples are representative of a medium-sized network comprising 30 nodes. The primary parameters for this medium-sized network are outlined in Table 1. A node density of one node per 400 square meters is achieved by evenly and randomly distributing the nodes throughout a 60 m × 60 m space.

For the experimental analysis, thirty nodes are considered in the simulation environment of network simulator-2 (NS-2) as show in Figure 4 and the data transfer is shown in Figure 5. Network latency is simply the time it takes to go from one end point to another over a distance of N hops. So, you have N segments (hops) with N-1 intermediate nodes. Each node has a delay (the cumulative effect of several things on that node, like queue delay, processing delays, etc), and each segment has a transit delay. Overall, that's 2N - 1 independent variables. So, it's seg1 +node1 +seg2 ... +node(N-1) +segN. The Figure 6 shows the Latency Vs Number of nodes.

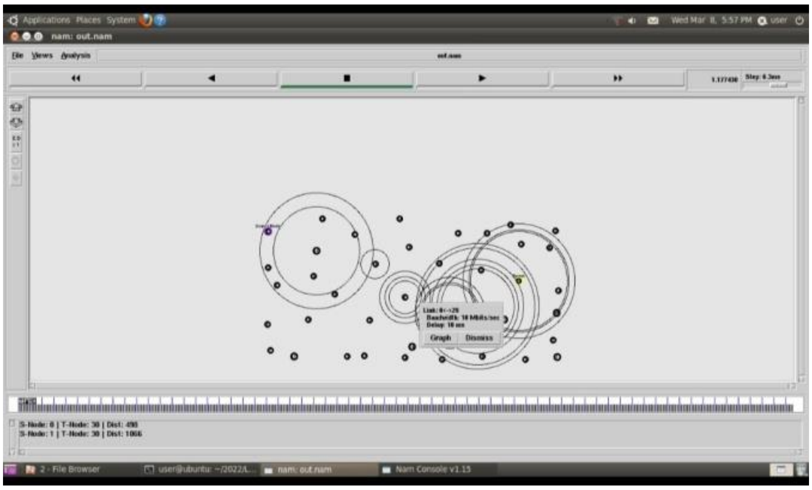

Figure 4. Simulation results of number of nodes

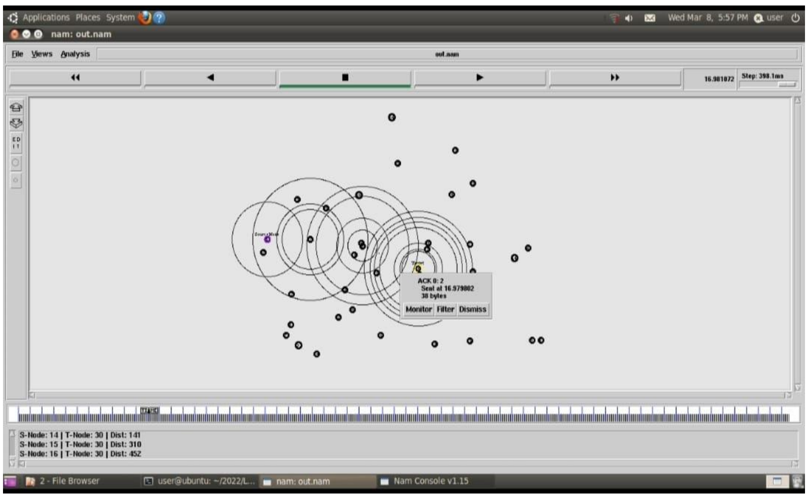

Figure 5. Simulation results of data transmission

Table 1. Simulation setup

Sl.no	Parameter	Value
1	Number of anchor nodes	3
2	Network Area	60m ×60m
3	Node density	1 per 400 m^2
4	Total number of sensor nodes	30
5	Communication range in ideal channel	20m

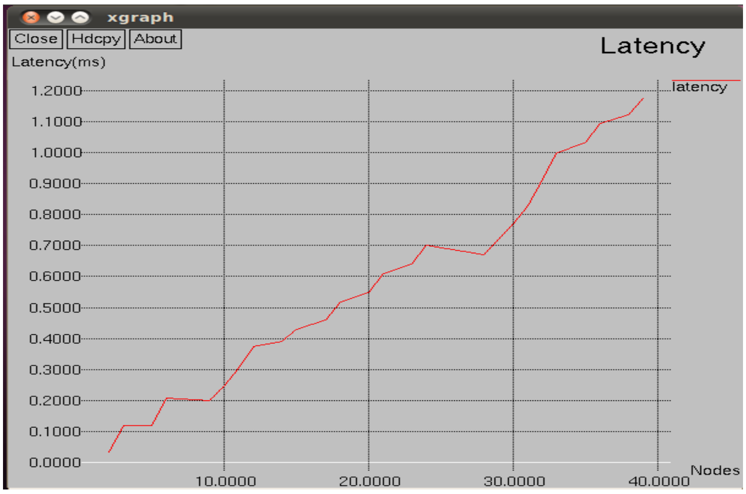

Figure 6. Latency vs number of nodes

4.1. Delay

The accuracy of sensor node localization can impact the delay in a wireless sensor network; however, there is no universally applicable formula that directly relates the delay to the number of nodes based solely on localization. The delay is influenced by various factors, including the communication protocol, network topology, data transmission rate, and processing capabilities of the sensor nodes. When sensor nodes are precisely localized as per the proposed algorithm, they can effectively communicate and route data, leading to reduced data packet losses and misdirection. This can result in lower packet retransmissions and improved overall network performance, leading to lower delays. Figure 7 represents the delay with respect to the number of nodes. Average it is less. When the number of nodes is doubled it keeps on increasing.

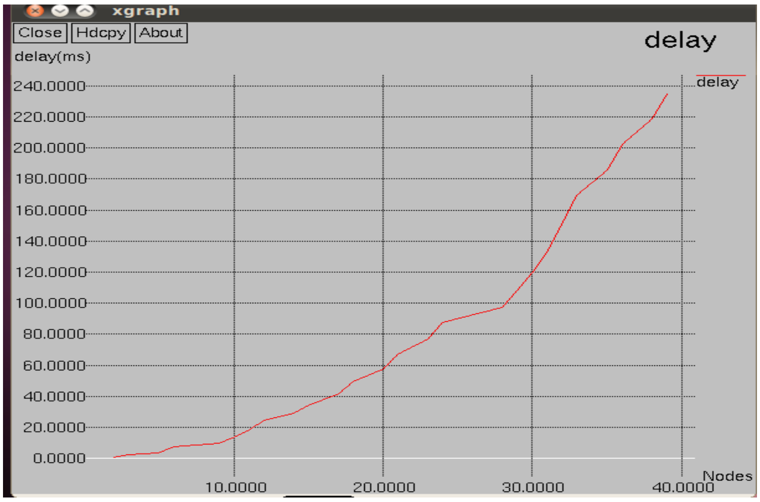

Figure 7. Delay vs number of nodes

4.2. Throughput

The location of sensor nodes' throughput is determined by the network's transmission characteristics and efficiency. In general, throughput is proportional to the transmission rate and the probability of successful transmission at the sensor node. It can be stated with the following formula:

Throughput = Transmission rate * Successful transmission probability.

Figure 8 demonstrates the throughput analysis with respect to the transmission rate. The localization process of sensor nodes impacts the transmission rate, which is determined by the frequency of data exchange among nodes during localization. When the localization algorithm necessitates frequent updates or location information exchanges among nodes, it can potentially affect the available bandwidth for other data transmission tasks. Consequently, the localization process may impose limitations on the transmission rate in such scenarios. The accuracy of localization can influence the successful transmission probability. When sensor nodes are precisely localized, they can establish improved communication links with each other, resulting in higher probabilities of successful transmission.

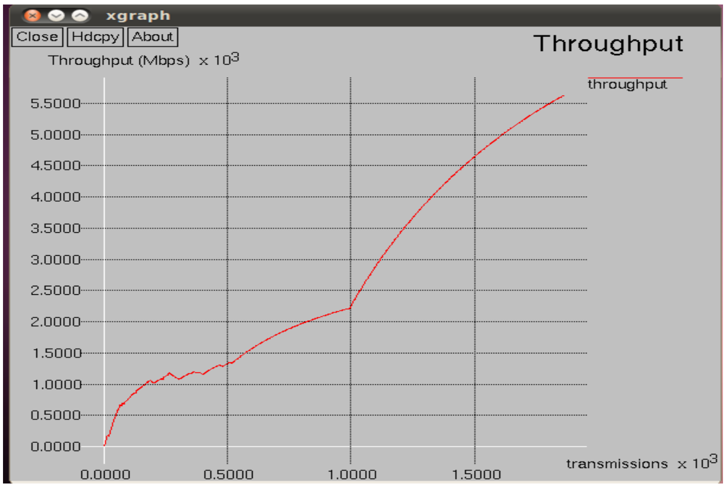

Figure 8. Throughput vs transmission rate

4.3. Error

The relationship between localization error and distance in sensor node localization can be expressed using various formulas, depending on the specific localization technique and the characteristics of the sensor network. The "error propagation model," which assumes that errors increase proportionally with the distance between the real and predicted locations of the sensor node, is a typical formula for modelling the error-distance relationship.

The error propagation model can be represented as:

Error = k * Distance + Bias

where:
- Error: The location error, representing the discrepancy between the estimated and true positions of the sensor node.
- Distance: The distance between the sensor node's real (true) position and its predicted position.
- k: A constant that represents the error propagation rate, indicating how fast the error increases with distance.
- Bias: A constant term that accounts for any systematic errors or inaccuracies in the localization process, which are not directly related to the distance.

Figure 9 illustrates the average localization errors for the algorithm in typical scenarios. Anchor nodes have a constant ratio of 10% to all other nodes. As the distance of location of sensor node increases, a noticeable trend emerges where the location errors of all relevant algorithms decrease consistently. Range-free localization methods, such as centroid-based or DV-hop localization as in Figure 10, based on how the distances are estimated or calculated in the network shows dominant error than GOA method because of the sensitivity of the bias.

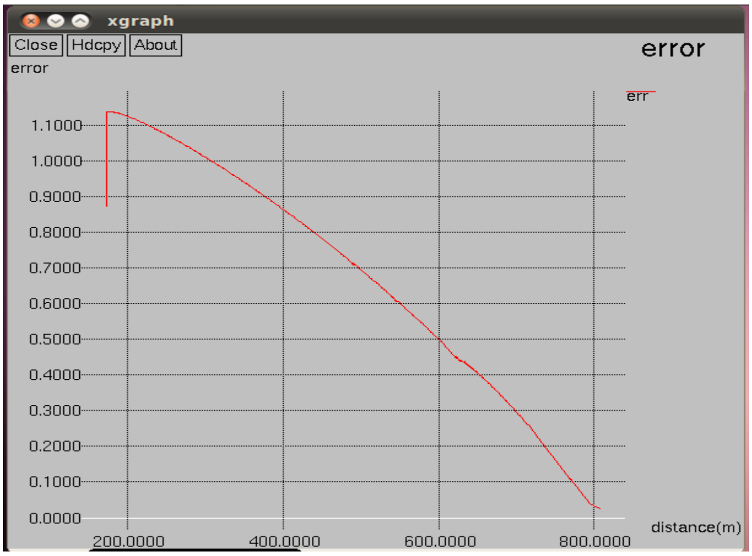

Figure 9. Error vs distance

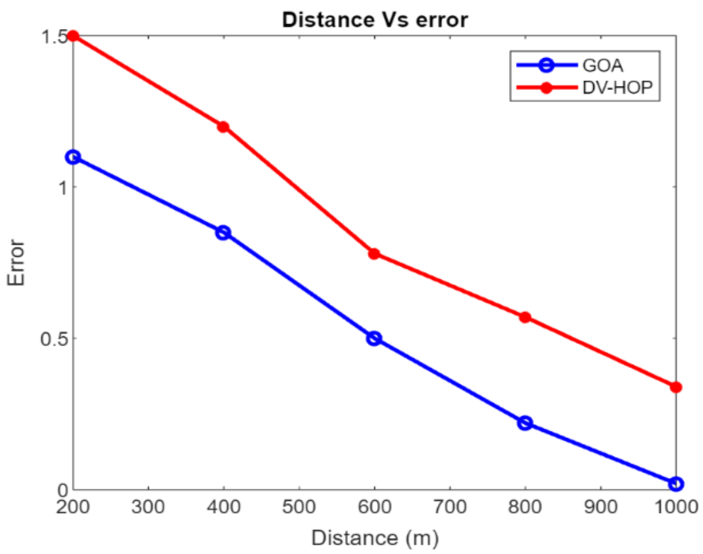

Figure 10. Comparison of error vs distance

5. CONCLUSIONS

Localization is a critical technique when things without GPS capabilities seek to identify their exact positions and coordinates, as well as those of other GPS-equipped objects, within the context of the IoT. IoT localization encounters unique challenges, and it's imperative for objects to achieve high levels of accuracy in determining their positions and locations. In most IoT applications, devices' ability to properly determine their positions and locations is critical because this information is required to convey vital data to the base station. The data's significance is frequently dependent on knowing the precise positions and locations of things within the IoT. Simultaneously, the efficacy of many routing methods is dependent on having precise knowledge of the objects' positions and locations. According to the test and implementation results, the GOA minimized the localization error of objects without GPS capability when compared to the DV-Hop and three-anchor approaches. Furthermore, the testing revealed that the number of GOA rounds correlates closely with an overall decrease in localization error. The obtained outcomes indicate that the proposed methodology effectively resolves and straightens out the three-anchor method's localization mistake, improving the degree of precision of item localization in the IoT.

The forthcoming pathways of this investigation encompass numerous pivotal realms for enhancement. To begin, practical experimentation and verification are essential to evaluate the efficacy of the GOA-DV-HL technique within genuine IoT and WSN settings. Modifying the algorithm to accommodate diverse networks, where sensor nodes possess varying capabilities, stands as another crucial focal point. Moreover, broadening the method to optimize mobile sensor nodes along with mobile sinks for adaptable route planning would greatly amplify its relevance in mobile networks. Integrating energy harvesting strategies to empower sensor nodes autonomously and weaving in multi-objective optimization to navigate through assorted conflicting aims could also elevate performance. Prospective studies might meld machine learning algorithms with GOA to further bolster the system's flexibility and efficacy. Enhancing the algorithm's scalability for extensive IoT networks and tackling security and privacy issues will be vital for instantaneous applications. Ultimately, delving into cross-layer optimization to boost energy efficiency, throughput, and network longevity will guarantee the algorithm can satisfy the demands of upcoming IoT and IIoT applications.

ACKNOWLEDGMENTS

All individuals acknowledged have provided their consent to be named in this section.

FUNDING INFORMATION

The authors did not receive support from any organization for the submitted work.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Janani Selvaraj	✓	✓	✓							✓	✓	✓	✓	
Hymlin Rose Sasijohn Gloryrajabai		✓			✓	✓		✓	✓	✓	✓	✓	✓	
Sivarathinabala Mariappan	✓		✓	✓			✓			✓	✓	✓	✓	
Backia Abinaya Antony Samy	✓	✓		✓					✓		✓	✓	✓	
Sudhakar Kalairishi		✓			✓			✓	✓					

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors have expressed no conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

ETHICAL APPROVAL

Not Applicable.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] B. Sun, Z. Cui, C. Dai, and W. Chen, "DV-hop localization algorithm with cuckoo search," *Sensor Letters*, vol. 12, no. 2, pp. 444–447, Feb. 2014, doi: 10.1166/sl.2014.3270.

[2] S. Zhang, M. J. Er, B. Zhang, and Y. Naderahmadian, "A novel heuristic algorithm for node localization in anisotropic wireless sensor networks with holes," *Signal Processing*, vol. 138, pp. 27–34, Sep. 2017, doi: 10.1016/j.sigpro.2017.03.010.

[3] S. M. Al Janabi and S. Kurnaz, "A new localization mechanism in IoT using grasshopper optimization algorithm and DVHOP algorithm," *Wireless Networks*, vol. 30, no. 6, pp. 5465–5485, 2024, doi: 10.1007/s11276-023-03247-2.

[4] X. Liu, S. Su, F. Han, Y. Liu, and Z. Pan, "A range-based secure localization algorithm for wireless sensor networks," *IEEE Sensors Journal*, vol. 19, no. 2, pp. 785–796, Jan. 2019, doi: 10.1109/JSEN.2018.2877306.

[5] M. Singh and P. M. Khilar, "A range free geometric technique for localization of wireless sensor network (WSN) based on controlled communication range," *Wireless Personal Communications*, vol. 94, no. 3, pp. 1359–1385, Jun. 2017, doi: 10.1007/s11277-016-3686-x.

[6] O. Cheikhrouhou, G. M. Bhatti, and R. Alroobaea, "A hybrid DV-hop algorithm using RSSI for localization in large-scale wireless sensor networks," *Sensors (Switzerland)*, vol. 18, no. 5, p. 1469, May 2018, doi: 10.3390/s18051469.

[7] T. Wang, H. Ding, H. Xiong, and L. Zheng, "A compensated multi-anchors TOF-Based localization algorithm for asynchronous wireless sensor networks," *IEEE Access*, vol. 7, pp. 64162–64176, 2019, doi: 10.1109/ACCESS.2019.2917505.

[8] T. Wang, H. Xiong, H. Ding, and L. Zheng, "TDOA-based joint synchronization and localization algorithm for asynchronous wireless sensor networks," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 3107–3124, May 2020, doi: 10.1109/TCOMM.2020.2973961.

[9] X. Kang, D. Wang, Y. Shao, M. Ma, and T. Zhang, "An efficient hybrid multi-station TDOA and single-station AOA localization method," *IEEE Transactions on Wireless Communications*, vol. 22, no. 8, pp. 5657–5670, Aug. 2023, doi: 10.1109/TWC.2023.3235753.

[10] S. Messous and H. Liouane, "Online sequential DV-hop localization algorithm for wireless sensor networks," *Mobile Information Systems*, vol. 2020, pp. 1–14, Oct. 2020, doi: 10.1155/2020/8195309.

[11] L. Jian Yin and M. Elhoseny, "A new distance vector-hop localization algorithm based on half-measure weighted centroid," *Mobile Information Systems*, vol. 2019, pp. 1–9, Jan. 2019, doi: 10.1155/2019/9892512.

[12] X. Li, K. Wang, B. Liu, J. Xiao, and S. Han, "An improved range-free location algorithm for industrial wireless sensor networks," *Eurasip Journal on Wireless Communications and Networking*, vol. 2020, no. 1, p. 81, Dec. 2020, doi: 10.1186/s13638-020-01698-1.

[13] D. Yu, T. Yuan, H. Qing, W. Xie, and P. Zhu, "Gray wolf optimizer with communication strategy based on DV-hop for nodes location of wireless sensor networks," *IEEE Access*, vol. 13, pp. 115307–115320, 2025, doi: 10.1109/ACCESS.2024.3445650.

[14] X. Yang and W. Zhang, "An improved DV-Hop localization algorithm based on hop distance and hops correction," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 11, no. 6, pp. 319–328, Jun. 2016, doi: 10.14257/ijmue.2016.11.6.28.

[15] J. Chen, W. Zhang, Z. Liu, R. Wang, and S. Zhang, "CWDV-hop: a hybrid localization algorithm with distance-weight DV-hop and CSO for wireless sensor networks," *IEEE Access*, vol. 9, pp. 380–399, 2021, doi: 10.1109/ACCESS.2020.3045555.

[16] S. Tomic and I. Mezei, "Improvements of DV-Hop localization algorithm for wireless sensor networks," *Telecommunication Systems*, vol. 61, no. 1, pp. 93–106, Jan. 2016, doi: 10.1007/s11235-015-0014-9.

[17] T. Wang, X. Wei, J. Fan, and T. Liang, "Adaptive jammer localization in wireless networks," *Computer Networks*, vol. 141, pp. 17–30, 2018, doi: 10.1016/j.comnet.2018.05.002.

[18] R. Kaviarasan, A. Ilavendhan, R. Rajakumar, Y. C. Hu, and C. C. Lo, "Improved grasshopper optimization with distance vector hop relay model for effective NLOS localization in VANETs," *International Journal of Communication Systems*, vol. 38, no. 7, May 2025, doi: 10.1002/dac.70063.

[19] V. Kanwar and A. Kumar, "DV-Hop localization methods for displaced sensor nodes in wireless sensor network using PSO," *Wireless Networks*, vol. 27, no. 1, pp. 91–102, Jan. 2021, doi: 10.1007/s11276-020-02446-5.

[20] R. Yuan, "Positioning of wireless sensor network under emergency communication environment," *Instrumentation Mesure Metrologie*, vol. 19, no. 4, pp. 273–279, Sep. 2020, doi: 10.18280/i2m.190404.

[21] A. Kaur, P. Kumar, and G. P. Gupta, "A weighted centroid localization algorithm for randomly deployed wireless sensor networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 31, no. 1, pp. 82–91, Jan. 2019, doi: 10.1016/j.jksuci.2017.01.007.

[22] X. Yang and W. Zhang, "An improved DV-Hop localization algorithm based on bat algorithm," *Cybernetics and Information Technologies*, vol. 16, no. 1, pp. 89–98, Mar. 2016, doi: 10.1515/cait-2016-0007.

[23] H. Puvitha, S. Palani, V. Vijayakumar, L. Ravi, and V. Subramaniyaswamy, "Investigation of multi-objective optimisation techniques to minimise the localisation error in wireless sensor networks," *International Journal of Grid and Utility Computing*, vol. 12, no. 1, pp. 33–42, 2021, doi: 10.1504/IJGUC.2021.112459.

[24] J. Mass-Sanchez, E. Ruiz-Ibarra, J. Cortez-González, A. Espinoza-Ruiz, and L. A. Castro, "Weighted hyperbolic DV-Hop positioning node localization algorithm in WSNs," *Wireless Personal Communications*, vol. 96, no. 4, pp. 5011–5033, Oct. 2017, doi: 10.1007/s11277-016-3727-5.

[25] M. Jiang *et al.*, "Improved DV-Hop localization algorithm based on anchor weight and distance compensation in wireless sensor network," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 12, pp. 167–176, Dec. 2016, doi: 10.14257/ijsip.2016.9.12.16.

[26] L. Gui, X. Zhang, Q. Ding, F. Shu, and A. Wei, "Reference anchor selection and global optimized solution for DV-Hop localization in wireless sensor networks," *Wireless Personal Communications*, vol. 96, no. 4, pp. 5995–6005, Oct. 2017, doi: 10.1007/s11277-017-4459-x.

[27] V. Gupta and B. Singh, "Study of range free centroid based localization algorithm and its improvement using particle swarm optimization for wireless sensor networks under log normal shadowing," *International Journal of Information Technology (Singapore)*, vol. 12, no. 3, pp. 975–981, Sep. 2020, doi: 10.1007/s41870-018-0201-5.

[28] M. Hosseinzadeh *et al.*, "A cluster-tree-based trusted routing algorithm using grasshopper optimization algorithm (GOA) in wireless sensor networks (WSNs)," *PLoS ONE*, vol. 18, no. 9 September, p. e0289173, Sep. 2023, doi: 10.1371/journal.pone.0289173.

BIOGRAPHIES OF AUTHORS

Dr. Janani Selvaraj is working as an Associate Professor in the Department of Electronics and Communication Engineering in Periyar Maniammai Institute of Science & Technology. She did her B.E. Degree (Electronics and Communication Engineering) in Kumaraguru College of Technology, Coimbatore and M.E Degree (Communication systems) in Sri Sivasubramaniya Nadar College of Engineering, Chennai. She pursued her Ph.D. from Annamalai University. She has more than 15 years of teaching experience, and her area of interest includes RF & Microwave Engineering, wireless communication, and networks. She presented many papers in various International Conferences, published 25 International Journals and 14 books. She is a member of ISTE, IEI, and IEEE. She can be contacted at email: jananiassociateprofessor2024@gmail.com.

Dr. Hymlin Rose Sasijohn Gloryrajabai graduated from Noorul Islam College of Engineering, Kumaracoil, Anna University, Chennai, in Electronics and Communication Engineering during the year 2007. She obtained her Master degree in Communication Systems from Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, Anna University, Chennai, during the year 2009. From 2009 to 2011. She has completed Ph.D. in Information and Communication, Anna University in the year 2021. She was working as an Assistant Professor in the department of Electronics and Communication Engineering at Kalasalingam University and Immanuel Arasar JJ College of Engineering, St. Joseph College of Engineering, Tamil Nadu. At present, she is working as an Assistant Professor in the Department of Electronics and Communication Engineering, R.M.D Engineering College, Chennai, Tamil Nadu, India. Her current area of research is security issues in wireless sensor networks. She can be contacted at email: hymlinrose@gmail.com.

Dr. Sivarathinabala Mariappan is working as an Associate Professor of Electronics and Communication Engineering at Velammal Institute of Technology, Tamil Nadu, India. She has been teaching in the Electronics and Communication field since 2007. She received her PhD from the College of Engineering, Guindy, Anna University, Chennai. Her research interest includes Computer vision, Image and video processing. She has published 25 international journals, 30 national and international conference papers and 3 patents. She received Best Researcher Award and SEEE Awards from the International Agencies. She is a member of IEEE Image Processing Society. She is acting as reviewer for various Elsevier, Springer, and IEEE journals. She acted as a resource person for various institutions. She can be contacted at email: msrb@velammalitech.edu.in.

Dr. Backia Abinaya Antony Samy is an esteemed Assistant Professor in the Department of Electronics and Communication Engineering at St. Joseph's College of Engineering and Technology, Thanjavur. With a profound background in Information and Communication Engineering, she earned her Ph.D., demonstrating a deep commitment to academic excellence. Dr. Abinaya has made significant contributions to the field through her prolific research endeavours. Her work has been published in renowned journals such as IEEE Explore, Scopus, and Web of Science, reflecting the high calibre of her scholarship. Additionally, she has actively participated in numerous IEEE conferences, enriching academic discourse and fostering collaboration within the scholarly community. She can be contacted at email: backiaabinaya@gmail.com.

Sudhakar Kalairishi is a student at the Periyar Maniammai Institute of Science and Technology. He is doing his B.Tech. (Hons.) degree in Electronics and Communication Engineering, specializing in robotics and industrial automation. He is the Secretary of the Robotics Club at PMIST, Vallam, and a Coordinator in E-Yantra PMIST. His areas of interest include robotics, mechatronics, and networks. He can be contacted at email: skrishi24153@gmail.com.