Hindawi
Journal of Computer Networks and Communications
Volume 2019, Article ID 1028723, 12 pages
https://doi.org/10.1155/2019/1028723

Research Article

Salp Swarm Algorithm for Node Localization in Wireless Sensor Networks

Huthaifa M. Kanoosh,[1] Essam Halim Houssein ,[2] and Mazen M. Selim[1]

[1]*Faculty of Computers and Informatics, Benha University, Banha, Egypt*
[2]*Faculty of Computers and Information, Minia University, Minya, Egypt*

Correspondence should be addressed to Essam Halim Houssein; essam.halim@mu.edu.eg

Received 27 July 2018; Revised 6 January 2019; Accepted 21 January 2019; Published 19 February 2019

Guest Editor: Noradin Ghadimi

Nodes localization in a wireless sensor network (WSN) aims for calculating the coordinates of unknown nodes with the assist of known nodes. The performance of a WSN can be greatly affected by the localization accuracy. In this paper, a node localization scheme is proposed based on a recent bioinspired algorithm called Salp Swarm Algorithm (SSA). The proposed algorithm is compared to well-known optimization algorithms, namely, particle swarm optimization (PSO), Butterfly optimization algorithm (BOA), firefly algorithm (FA), and grey wolf optimizer (GWO) under different WSN deployments. The simulation results show that the proposed localization algorithm is better than the other algorithms in terms of mean localization error, computing time, and the number of localized nodes.

1. Introduction

Wireless sensor networks (WSNs) are networks that consist of a few autonomous sensor nodes that are distributed in a specific area for cooperatively sensing their environment. In the last decade, WSNs have been employed in many real-life applications such as healthcare, home automation, traffic surveillance, and environmental monitoring [1].

Accurate node localization is a critical issue in WSNs. Localization problem in WSNs means calculating the positions of unknown sensor nodes. In many environments where the sensor nodes of a WSN are deployed, people may not be able to go and fix these nodes. In these environments, the sensor nodes are usually randomly scattered in random locations; therefore, the sensor nodes usually take random positions. On the other hand, in many applications, the information collected by the sensor nodes of a WSN may be useless if the positions of the sensor nodes that gathered the information are unknown. This emphasizes the need for an accurate node localization scheme [2].

In order to localize the sensor nodes of a WSN, a GPS can be attached to each sensor during the network deployment. Then, the GPS can be used to find the coordinates of the sensor nodes. However, using GPS for localizing sensor nodes is undesirable and impractical solution because of many reasons such as cost, inaccessibility, sensor nodes may be deployed indoors, and climatic conditions may disturb the GPS reception ability [2]. An alternative approach is to attach GPS to some nodes which are called anchor nodes or beacon nodes. Thus, the positions of these nodes with attached GPS are known after deploying the nodes of the wireless sensor network. Using the known locations of anchor nodes, localization algorithms can be employed to estimate the positions of the unknown nodes [2]. There are two classes of non-GPS based localization algorithms, namely, rang-free and range-based algorithms [3]. Angle-based or point-to-point distance estimation among the sensor nodes is used with rang-based localization algorithms. In these algorithms, the positions of sensor nodes are calculated by the assist of anchor's trilateration [3]. Unlike range-based localization algorithms, range-free localization algorithms do not need range information to estimate the positions of the unknown nodes. It only depends on the topological information. Compared to range-free localization algorithms, range-based localization algorithms can achieve higher localization accuracy but at the expense of cost [4].

Recently, node localization in WSNs is handled as a multimodal and multidimensional optimization problem that can be solved using population-based stochastic approaches. In the literature, many metaheuristic algorithms have been employed to solve the localization problem in WSNs. These algorithms have succeeded in reducing the localization error dramatically. These algorithms attempt to solve an optimization problem by trial and error in which the feasible solutions are processed, and the nearest optimal solution is identified. Currently, various optimization algorithms such as genetic algorithm (GA), cuckoo search (CS), gravitational search algorithm (GSA), butterfly optimization algorithm (BOA), particle swarm optimization (PSO), artificial bee colony (ABC), etc. have been employed effectively in specifying the positions of the unknown nodes in WSNs [5].

The main contribution of this paper is using the Salp Swarm Algorithm (SSA) for the first time ever to localize the nodes of WSNs. The performance of the proposed SSA-based localization algorithm is analyzed and compared with particle swarm optimization (PSO), butterfly optimization algorithm (BOA), firefly algorithm (FA), and grey wolf optimizer (GWO) algorithms. The results have shown that the SSA-based localization algorithm is better than the previously mentioned localization algorithms in terms of computing time, number of localized nodes, and localization accuracy.

The remaining sections of the paper are organized as follows: Section 2 covers some of the research efforts which have been done in the field. Section 3 presents a brief description for the different swam algorithms employed in this work. Section 4 introduces the proposed SSA-based localization algorithm. Section 5 includes the conducted experiments and results analysis. Finally, the paper is concluded in Section 6.

2. Literature Review

In the last years, many optimization algorithms have been employed for addressing the problem of node localization in WSNs [6]. In this section, some of the recent relevant works are covered and briefly described.

Low et al. have introduced a low-cost localization system that depends on the measurements obtained from a pedometer and communication ranging among adjacent nodes. The proposed system does not require good network connectivity and presents good performance in sparse networks. A probability-based algorithm that needs a nonlinear optimization problem solving is employed to provide the localization information. Moreover, the particle swarm optimization (PSO) has been employed to determine the optimum location of the sensor nodes. Experimental results have proved that the proposed system has a good performance [7].

Manjarres et al. have presented a hybrid node localization algorithm based on Harmony Search (HS) algorithm with a local search procedure. The main objective of the proposed algorithm is to address the localization problem and to distribute its burdens over an iterative process. Additionally, the proposed algorithm employs certain connectivity-based geometrical constraints to decrease the area in which each node can be located. The simulation results have verified that the proposed algorithm is better than another simulated annealing- (SA-) based localization algorithm in terms of both localization error and computational cost [8].

Li et al. have suggested a self-adaptive artificial bee colony (SAABC) node localization algorithm that considers the whole effects resulting from employing dynamic topology. The proposed algorithm provides good performance in WSNs with both random distributing nodes and dynamic topology. Additionally, the obtained simulation results have shown that the proposed localization algorithm provides better node localization precision and precision stability compared to the DV-Hop algorithm without the need for extra devices or overhead in communication [9].

Tamizharasi et al. have proposed a novel hybrid node localization algorithm based on bacterial foraging algorithm (BFA) and particle swarm optimization (PSO). The main design objectives of the proposed algorithm are to enhance the efficiency and accuracy of BFA and to avoid getting stuck in a local extreme. In the proposed algorithm, PSO is merged into the chemotaxis of BFA to speed up the convergence rate. Moreover, the global search ability is improved by proposing the elimination probability in elimination-dispersion based on the energy of bacteria. The obtained simulation results have verified that the proposed hybrid algorithm outperforms the BFA [10].

Tang et al. have proposed a sensor nodes localization algorithm, which depends on a new intelligent optimization algorithm called plant growth simulation algorithm (PGSA) that simulates the growth of plants. In their work, they proposed inserting the plant root of adaptive backlight function into the original PGSA for improving convergence time and localization precision. The obtained simulation results have verified that the proposed algorithm is better than the simulated annealing algorithm (SAA) in terms of localization accuracy and computing time [11].

Jegede and Ferens have employed the genetic algorithm (GA) for learning the environmental issues within a WSN for effectively localizing its sensor nodes. For every coordinate in the grid network area, given random perturbations of received signal strength (RSS), GA would be able to learn the environment and to decrease the possible errors associated with the RSSI measurement taken for each coordinate. The conducted simulation shows that GA can reasonably localize sensor nodes using the coordinates of three anchors [12].

Goyal and Patterh have proposed a cuckoo search- (CS-) based node localization algorithm for estimating the coordinates of sensor nodes in WSNs. In the proposed algorithm, no weight coefficient is employed for controlling the global search ability. The conducted simulation has shown that the proposed localization algorithm is better than the particle swarm optimization (PSO) and various variants of biogeography-based optimization (BBO) in terms of localization accuracy [13].

Dan and Xian-bin have presented a distributed two-phase PSO algorithm for efficiently and precisely localizing the sensor nodes in addition to solving the flip ambiguity problem. In the first phase, the initial search space is defined using the bounding box method. In the second phase, a refinement process is performed for correcting the error resulting from the flip ambiguity. Additionally, the proposed algorithm attempted localizing sensor nodes that have only two references or three near-collinear references. The conducted simulation has proved the effectiveness of the proposed algorithm [14].

Krishnaprabha and Gopakumar have proposed a node localization algorithm based on gravitational search algorithm (GSA). In the proposed work, node localization in WSNs is formulated as a nonlinear optimization problem. Also, the proposed algorithm tried to handle the flip ambiguity problem and to localize the sensor nodes that collinear anchor nodes through the refinement phase. The obtained simulation results have shown that the proposed localization algorithm has good performance [15].

Peng and Li have focused on range-free localization as a cost-effective alternative compared to range-based approaches. However, they noticed that range-free localization suffers from higher localization error compared to the range-based algorithms. In order to deal with this problem, they presented an improved version for DV-Hop, which is a popular rang-free approach that depends on hop-distance estimation. The improvement in the DV-Hop algorithm is performed based on a genetic algorithm. Simulation results have shown that the proposed localization algorithm has better localization accuracy compared to other localization algorithms [16].

Sai et al. have presented a hybrid node localization algorithm in WSNs, which depends on the measurements of the received signal strength (RSS) and parallel firefly algorithm (PFA). Taking into consideration the distance factor, the proposed algorithm transforms the node localization problem in WSN into a nonlinear unconstrained optimization problem that is defined by an enhanced objective function. In the proposed algorithm, PFA estimates the coordinates of sensor nodes using the distances between a sensor and a few numbers of its 1-hop neighbors. Simulation results have shown that the proposed approach is better than PSO, GA, PFA, and RSS in terms of localization accuracy [17].

Sivakumar and Venkatesan have provided two-phase node localization algorithm in WSNs. In the first phase, the positions of sensor nodes are roughly estimated using a range-free localization method called mobile anchor positioning (MAP). In the second phase, the proposed algorithm attempts to reduce the localization error by employing a certain metaheuristic approach. In their work, to perform the second phase, they employed bat optimization algorithm (BOA), modified cuckoo search (MCS), and firefly optimization algorithm (FOA) resulting in three localization algorithms namely, BOA-MAP, MCS-MAP, and FOA-MAP. The experimental results have shown that FOA-MAP is better than both BOA-MAP and MCS-MAP in terms of root mean square error (RMSE) [18].

Sun et al. have proposed a multiobjective node localization algorithm based on multiobjective particle swarm optimization, called multiobjective particle swarm optimization localization algorithm (MOPSOLA). The multiobjective functions include the space distance and the geometric topology constraints. In the proposed algorithm, the size of the archive remains limited using a dynamic method. Simulation results have shown that the proposed algorithm has achieved significant enhancements in terms of localization accuracy and convergence rate [19].

Arsic et al. have proposed a node localization algorithm based on fireworks algorithm (FWA). The proposed algorithm provides optimal results in case of no ranging errors and provides good results in case of ranging errors. Moreover, they proposed an enhanced fireworks algorithm (EFWA), which achieved better results compared to FWA. Also, the proposed localization algorithm outperforms the existing PSO-based localization algorithm [20].

Shieh et al. have compared several well-known heuristics such as genetic algorithm (GA) and particle swarm optimization (PSO) to more recent methods such as grey wolf optimizer (GWO), firefly algorithm (FA), and brain storm optimization (BSO) algorithms in terms of sensor nodes localization accuracy. Also, they proposed an enhancement in the localization algorithms to increase the number of localized nodes. The improved algorithms have been compared to the original ones in terms of the number of localized nodes and execution time in different network deployments [21].

Cheng and Xia have proposed a cuckoo search (CS) algorithm-based node localization algorithm. In the proposed method, the step size has been modified to obtain a global optimal solution in a short time. Also, the candidate solutions' fitness is used for constructing mutation probability to avoid local convergence. The performance of the proposed algorithm has been evaluated using different anchor density, node density, and communication range in terms of average localization error and localization success ratio. The simulation results have proved that the proposed algorithm is better than the standard CS and PSO in terms of average localization error and convergence time [22].

Daely and Shin have proposed a node localization algorithm based on dragonfly algorithm (DA) optimization algorithm. The proposed localization algorithm was designed to determine the positions of the nodes which are randomly distributed in a specific area. The simulation results proved that the proposed DA based algorithm is better than PSO in terms of localization accuracy [23].

Nguyen et al. have employed the multiobjective firefly algorithm to estimate the coordinates of sensor nodes in WSNs. The used objective functions depend on two criteria, namely, the nodes' distance constraint and geometric topology constraint. The simulation results have shown the superiority of the proposed algorithm compared to well-known localization methods in terms of localization accuracy and convergence rate [24].

Arora and Singh have used the butterfly optimization algorithm to localize the sensor nodes in WSNs. The proposed localization algorithm has been validated using different numbers of nodes with distance measurements corrupted through the Gaussian noise. The simulation results have shown that the proposed localization algorithm is better than several well-known localization algorithms including particle swarm optimization (PSO) and firefly algorithm (FA) in terms of localization accuracy [25].

Ahmed et al. have presented a node localization algorithm based on whale optimization algorithm (WOA) whose main objective is to localize sensor nodes in WSNs accurately [6]. Also, Sujatha and Siddappa have proposed a hybrid localization algorithm based on dynamic weight particle swarm (DWPSO) and differential evolution (DE) algorithms. The authors observed that decreasing the square error of estimated and measured distance can improve the localization accuracy. The obtained simulation results proved that the proposed localization algorithm is better than the linearization method (LM) in terms of localization accuracy and performance stability [26].

Kaur and Arora have compared the performance of several bioinspired algorithms including flower pollination algorithm (FPA), firefly algorithm (FA), grey wolf optimization (GWO) and particle swarm optimization (PSO) in localizing the sensor nodes of WSNs. The performance of the different algorithms has been evaluated in terms of several performance metrics including localization accuracy, computing time, and several localized nodes. The simulation results have shown the superiority of FPA compared to the other algorithms in terms of localization accuracy [27].

To the best of our knowledge, the Salp Swarm Algorithm (SSA) algorithm was never used for the localization problem in WSNs so far. Therefore, the main objective of this paper is to employ the SSA algorithm for handling the localization problem in WSNs and is to evaluate its performance against several well-known swarm intelligence algorithms. The basic ideas behind the SSA and other swarm algorithms are given in the next section.

3. Swarm Intelligence Algorithms

Swarm intelligence (SI) is a relatively new interdisciplinary field of research, which has gained huge popularity in these days. Algorithms belonging to this domain draw inspiration from the collective intelligence emerging from the behavior of a group of social insects (like bees, termites, and wasps). It has successfully been applied to several real-world optimization problems [28–30]. In this section, we review some of these algorithms that are employed in this paper in order to localize the nodes of WSNs.

3.1. Particle Swarm Optimization. Particle swarm optimization (PSO) is a swarm optimization algorithm proposed by Eberhart and Kennedy in 1995. It is inspired by the collective behavior of bird flocking and fish schooling. It employs several particles that simulate a swarm population moving around in the search space to find the best solution. Each particle provides a candidate solution for the problem and is usually represented as a point in D-dimensional space. The position of each particle is represented as a vector $x_i = (x_{i1} \cdot x_{i2} \cdot \ldots \cdot x_{i\,D})$. Particles move in the search space to find optimal solutions, and each particle has a velocity represented as a vector $v_i = (v_{i1} \cdot v_{i2} \cdot \ldots \cdot v_{i\,D})$. During movement, the position of each particle is updated based on the best position that has been achieved by the particle itself (*pbest*) and the best position that has been achieved by all the particles so far (*gbest*). The velocity and the position of each particle are updated as shown below [31]:

$$x_{\mathrm{id}}^{t+1} = x_{\mathrm{id}}^{t} + v_{\mathrm{id}}^{t+1}, \quad (1)$$

$$v_{\mathrm{id}}^{t+1} = w * v_{\mathrm{id}}^{t} + c_1 * r_1 * \left(p_{\mathrm{id}} - x_{\mathrm{id}}^{t}\right) + c_2 * r_2 * \left(p_{\mathrm{gd}} - x_{\mathrm{id}}^{t}\right), \quad (2)$$

where t refers to the iteration number, w is inertia weight that aims at determining the effect of previous velocities on current velocity, c_1 and c_2 are acceleration constants, r_1 and r_2 are random variables whose values are normally distributed in [0, 1], and p_{id} and p_{gd} indicate the elements of *pbest* and *gbest* in the d^{th} dimension, respectively. The velocity is limited by a predefined maximum velocity $v_{\max}$.

3.2. Butterfly Optimization Algorithm. Butterfly optimization algorithm (BOA) is a swarm intelligence algorithm that is recently proposed by Arora and Singh [32]. It is inspired from the food-foraging behavior of butterflies. Biologically, butterflies find the source of food using sense receptors. The role of these sense receptors, also called chemoreceptors, is to sense fragrance/smell [33].

In BOA, butterflies are the search agents that perform optimization. Each butterfly emits fragrance with some intensity. This fragrance is propagated and sensed by other butterflies in the region. The fragrance emitted by butterfly is correlated with the butterfly's fitness. This means that the fragrance of a butterfly changes according to its current location [32]. When a butterfly is able to sense fragrance from any other butterfly that is larger than its fragrance, it will move toward the latter, and this phase is called global search. On the other side, when a butterfly cannot sense fragrance from other butterflies that is larger than its local fragrance, it will move randomly, and this phase is called local search [34]. In BOA, the fragrance is defined as a function of the physical intensity of stimulus [34] as shown below:

$$f_i = cI^a, \quad (3)$$

where f_i is the perceived magnitude of fragrance, c is the sensory modality, I is the stimulus intensity, and a is the power exponent dependent on modality, which accounts varying degree of absorption. As mentioned before, there are two phases in the BOA algorithm, namely, local search phase and global search phase. In global search phase, the butterfly moves one step toward the best butterfly/solution g^* which can be formulated as

$$x_i^{t+1} = x_i^t + \left(r^2 \times g^* - x_i^t\right) \times f_i, \quad (4)$$

where x_i^t is the solution vector x_i of i^{th} butterfly in iteration number t, g^* is the best solution found in the current stage,

f_i is the fragrance of i^{th} butterfly, and r is a random number in $[0, 1]$. Local search phase is formulated as

$$x_i^{t+1} = x_i^t + \left(r^2 \times x_k^t - x_j^t\right) \times f_i, \quad (5)$$

where x_j^t and x_k^t are j^{th} and k^{th} butterflies chosen randomly from the solution space. Equation (5) is considered a local random walk if and only if x_j^t and x_k^t belongs to the same subswarm, and r is a random number in $[0, 1]$. Search for food and mating partner by butterflies can occur at both local and global scale; therefore, a switch probability p is employed in BOA to switch between common global search and intensive local search.

3.3. Firefly Algorithm. Firefly algorithm (FA) is a nature-inspired algorithm proposed by Yang and He [35]. It mimics the social behaviors and flashing patterns of fireflies. FA depends on the next three idealized rules [36]:

(i) Fireflies are unisex which means that a firefly can get attracted to any other firefly regardless their sex.

(ii) The attractiveness of fireflies is directly proportional to their brightness. Thus, for any two flashing fireflies, the firefly with less brightness moves toward the one with higher brightness. If there is no a brighter firefly than a particular firefly, the latter moves randomly.

(iii) The brightness of a firefly is calculated using an objective function.

A firefly's attractiveness is proportional to the light intensity visualized by other fireflies in the region; therefore, the relationship between the attractiveness β and the distance r can be formulated as

$$\beta = \beta_0 e^{-\gamma r^2}, \quad (6)$$

where β_0 is the brightness at distance $r = 0$ and γ is the light absorption coefficient. The movement of a firefly i toward a more attractive (brighter) firefly j is represented as

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \alpha_t \epsilon_i^t, \quad (7)$$

where r_{ij} is the distance between the fireflies i and j that is calculated using the Euclidean norm, the second term is due to the attraction, the third term is a randomization with α_t being the randomization parameter, and ϵ_i^t is a vector that includes random numbers.

3.4. Grey Wolf Optimizer. Grey wolf optimizer (GWO) is a recent swarm intelligence algorithm inspired by the grey wolf community. It is developed by Mirjalili et al. in 2014. Grey wolf is a very dangerous creature which belongs to the Canidae family. Grey wolves usually live in packs that consist of 5 to 12 wolves. Each group has social dominance hierarchy: alpha, beta, and omega, in order. The alphas are a male and female which represent the leaders of the group. The betas are the second level of management hierarchy. The omegas are the final level in the hierarchy [37].

In order to mathematically model the social hierarchy of wolves in GWO, the fittest solution is referred to as the alpha (α). Consequently, the second and third best solutions are beta (β) and delta (δ), respectively. The rest of candidate solutions are omega (ω) [35]. The mathematical model of the encircling behavior is represented as follows [35]:

$$\begin{aligned} \vec{D} &= |\vec{C}.\vec{X}p\ (t) - \vec{X}(t)|, \\ \vec{X}(t+1) &= |\vec{X}p(t) - \vec{A}.\vec{D}|, \end{aligned} \quad (8)$$

where t indicates the current iteration, $\vec{C} = 2.\vec{r}_2$, $\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a}$, $\vec{X}m$ is the position vector of the wolf, r_1 and r_2 are random vectors in [0, 1] and linearly varies from 2 to 1, C and A are coefficient vectors, and Xp is the position vector of the prey. During the optimization, ω wolves update their positions around $\alpha.\beta.$ and δ. Therefore, the ω wolves can reposition with respect to α, β, and δ as shown below [35]:

$$\begin{aligned} \vec{D}\alpha &= \left|\vec{C}_1.\vec{X}\alpha - \vec{X}\right|.\vec{D}\beta = \left|\vec{C}_2.\vec{X}\beta - \vec{X}\right|.\vec{D}\delta \\ &= \left|\vec{C}_3.\vec{X}\delta - \vec{X}\right|, \\ \vec{X}1 &= \vec{X}\alpha - \vec{A}_1.(\vec{D}\alpha).\vec{X}_2 = \vec{X}\beta - \vec{A}_2.(\vec{D}\beta).\vec{X}_3 \\ &= \vec{X}\delta - \vec{A}_3.(\vec{D}\delta), \\ \vec{X}(t+1) &= \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}. \end{aligned} \quad (9)$$

With these equations, a search agent updates its position according to $\alpha.\ \beta.$ and δ in an n-dimensional search space. In addition, the final position would be in a random place within a circle which is defined by the positions of $\alpha.\beta.$ and δ in the search space [37].

3.5. Salp Swarm Algorithm. Salps are part of Salpidae family with the limpid cylinder design body. They look like jellyfishes in texture and movement. The shape of a Salp is shown in Figure 1(a). The water is pushed Salps bodies to move forward [38]. Generally, the biological research about Salps is still in its early stages because their living environments are hardly accessible, and it is very difficult to keep them in laboratory environment. Salps swarming attitude is the main inspiration to build Salp swarm algorithm [39]. Salps compose a swarm in profound oceans which is called Salp chain. This chain is illustrated in Figure 1(b). This chain can help to achieve better locomotion during the foraging process [40].

Originally, the Salps population is divided into two groups to formulate the mathematical model for Salp chains: head and followers. The head position is at the beginning of the chain while the rest of the chain is referred to as the followers [41].

The Salps location is determined likewise swarm-based methods, by an n-dimensional search area via considering

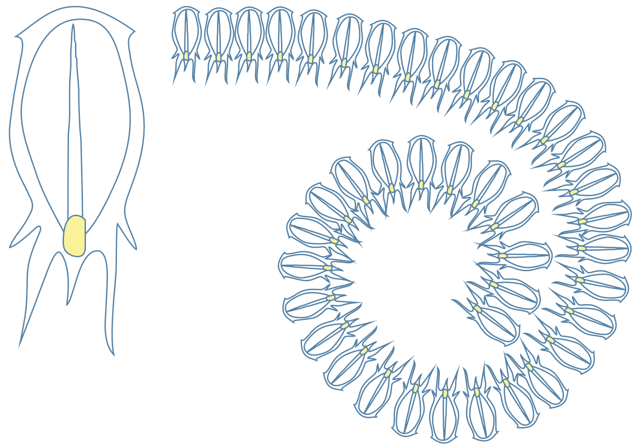

Figure 1: (a) Individual Salp. (b) Salps chain [37].

the number of variables inside the presented problem represented by n. Accordingly, a two-dimensional matrix described as x will reservation the position of all Salps. It is supposed too, in search space, there is "F" which is a food source as the target of the swarm. The following equation is suggested to upgrade the leader location [37]:

$$x_j^1 = \begin{cases} F_j + c_1\left(\left(\mathrm{ub}_j - \mathrm{lb}_j\right)c_2 + \mathrm{lb}_j\right), & c_3 \geq 0, \\ F_j - c_1\left(\left(\mathrm{ub}_j - \mathrm{lb}_j\right)c_2 + \mathrm{lb}_j\right), & c_3 < 0, \end{cases} \tag{10}$$

where x_j^1 shows the position of the first Salp (leader) in the j^{th} dimension, F_j is the position of the food source in the j^{th} dimension, ub_j indicates the upper bound of j^{th} dimension, lb_j indicates the lower bound of j^{th} dimension, and c_1 , c_{the2} , and c_3 are random numbers.

Equation (10) shows that the leader only updates its position with respect to the food source. The coefficient c_1 is the most important parameter in SSA because it balances exploration and exploitation defined as follows [37]:

$$c_1 = 2e^{-(4l/L)^2}, \tag{11}$$

where l is the current iteration and L is the maximum number of iterations.

The parameters c_2 and c_3 are random numbers uniformly generated in the interval of [0, 1]. In fact, they dictate if the next position in j^{th} dimension should be towards positive infinity or negative infinity as well as the step size. To update the position of the followers, the following equations is utilized (Newton's law of motion) [37]:

$$x_j^i = \frac{1}{2}at^2 + v_0 t, \tag{12}$$

where $i \geq 2$, x_j^i shows the position of i^{th} follower Salp in j^{th} dimension, t is time, v_0 is the initial speed, and $a = v_{\text{final}}/v_0$ where $v = x - x_0/t$.

Because the time in optimization is an iteration, the discrepancy between iterations is equal to 1 and considering $v_0 = 0$, this equation can be expressed as follows [37]:

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right), \tag{13}$$

where $i \geq 2$ and x_j^i shows the position of i^{th} follower Salp in j^{th} dimension. Using equations (1) and (4), the Salp chains can be simulated. Figure 2 shows the pseudocode to implement the SSA algorithm.

4. Formulation of WSN Localization Problem

WSN node localization problem is formulated using the single hop range-based distribution technique to estimate the position of the unknown node coordinates (X, Y) with the aid of anchor nodes (position of known nodes) coordinates (x, y). Anchor nodes are provided with a GPS device, so it has the capability of automatically determining its position. Most of the nodes in the WSN are not equipped with GPS due to high cost. To measure the coordinates of N unknown nodes, the procedure followed is given below.

Step 1. Randomly initialize the N unknown nodes and M anchor nodes within the communication range (R). Anchor nodes measure their position and communicate their coordinates to their neighbors. For all iterations, the node which settles at the end is termed as the reference node, and this node will act as anchor node.

Step 2. Three or more anchor nodes within the communication range of a node are considered as a localized node.

Step 3. Let (x, y) be the coordinates of the target node to be determined and d_i be the distance between the target node and the i^{th} anchor node.

$$d_i = \sqrt{\left(x - x_i\right)^2 + \left(y - y_i\right)^2}. \tag{14}$$

Step 4. The optimization problem is formulated to minimize the error of the localization problem. The objective function for the localization problem is formulated as

```
Initialize the Salp population x_i (i = 1, 2, ..., n) considering u_b and l_b
While (end condition is not satisfied)
Calculate the fitness of each search agent (salp)
F = the best search agent
Update c_1 by equation (2)
    For each Salp (x_i)
        If (i == 1)
            Update the position of the leading Salp by equation (1)
        Else
            Update the position of the follower Salp by equation (4)
        End
    End
    Amend the Salps based on the upper and lower bounds of variables
End
Return F
```

Figure 2: Pseudocode of the SSA algorithm.

$$f(x, y) = \min\left(\sum_{i=1}^{M}\left(\sqrt{(x - x_i)^2 + (y - y_i)^2}\right)^2\right), \quad (15)$$

where M is the anchor nodes within the transmission range (R) of the target node.

Step 5. All target localized nodes (N_L) are determined, the whole localization error is calculated as the mean of the square of distances of the estimated coordinates node (x_i, y_i) and the actual node coordinates (X_i, Y_i), for $i = 1, 2, \ldots, N_L$:

$$E_L = \frac{1}{N_L}\sum_{i=1}^{L}\left(\sqrt{(x_i - X_i)^2 + (y_i - Y_i)^2}\right). \quad (16)$$

The performance of SSA algorithm evaluated using E_L and the number of nonlocalized nodes N_{NL}, where $N_{NL} = [N - N_L]$.

Step 6. Repeat the steps 2–5 until all unknown/target nodes get localized or no more nodes can be localized.

5. Experimental Analysis

In this section, the proposed WSN localization approach is evaluated under different scenarios, and its performance is compared to four other swarm-based algorithms (PSO, BOA, FA, and GWO) in terms of localization accuracy and computing time. The computations of the different algorithms are performed using MATLAB R2012b on a machine of Intel Core i7 CPU, 4 GB RAM, and Windows7 operating system. The parameters' values of the deployment area are shown in Table 1.

For BOA, the sensory modality c is set 0.01, whereas the initial value of power exponent a is set to 0.1 [25]. For PSO, initial values of $\omega = 0.7$ and $c_1 = c_2 = 1.494$ were recommended for faster convergence after experimental tests [25]. For FA, the randomization parameter α is set to 0.25, the absorption coefficient γ is set to 1.0, and the initial attractiveness parameter β is set to 1. For GWO, the parameter a linearly decreases in the interval of [2 to 0] and the C parameter linearly increases from 0 to 2 [6]. Finally, for SSA, c_1 is calculated using equation (2), whereas c_2 = 0.7 and c_3 = 0.3 and c_2, c_3 are random numbers uniformly generated from the interval [0, 1].

Table 1: Parameters setting of simulation environment.

Parameters	Values
Sensor nodes	Varies on $\sum_{i=1}^{6} i * 25$
Anchor nodes	Varies on increment $i = i + 5$
Node transmission range (R)	30 m
Deployment area	100 m × 100 m
Maximum number of iterations	100

5.1. Sensor Nodes Localization Using SSA. In all conducted experiments, the coordinates of central nodes (N) and destination nodes (M) are randomly configured during the construction of the deployment area. The deployment area includes three types of nodes: anchor nodes whose known position, target nodes whose unknown position, and localized nodes whose positions are already estimated. In this section, the performance of the SSA-based localization algorithm is evaluated under different scenarios using different numbers of target nodes and different numbers of anchors as shown in Figure 3.

5.2. Comparison among Different Localization Algorithms. In this section, SSA and the other swarm algorithms have been evaluated under different scenarios in terms of localization error, computing time, and number of localized nodes. The obtained results of the different algorithms are shown in Table 2.

Under the different scenarios (number of nodes/number of anchors), it is noticed that for all the localization algorithms, increasing the number of iterations increases both of the number of localized nodes and the computing time while reduces the localization error. This notice is rational because

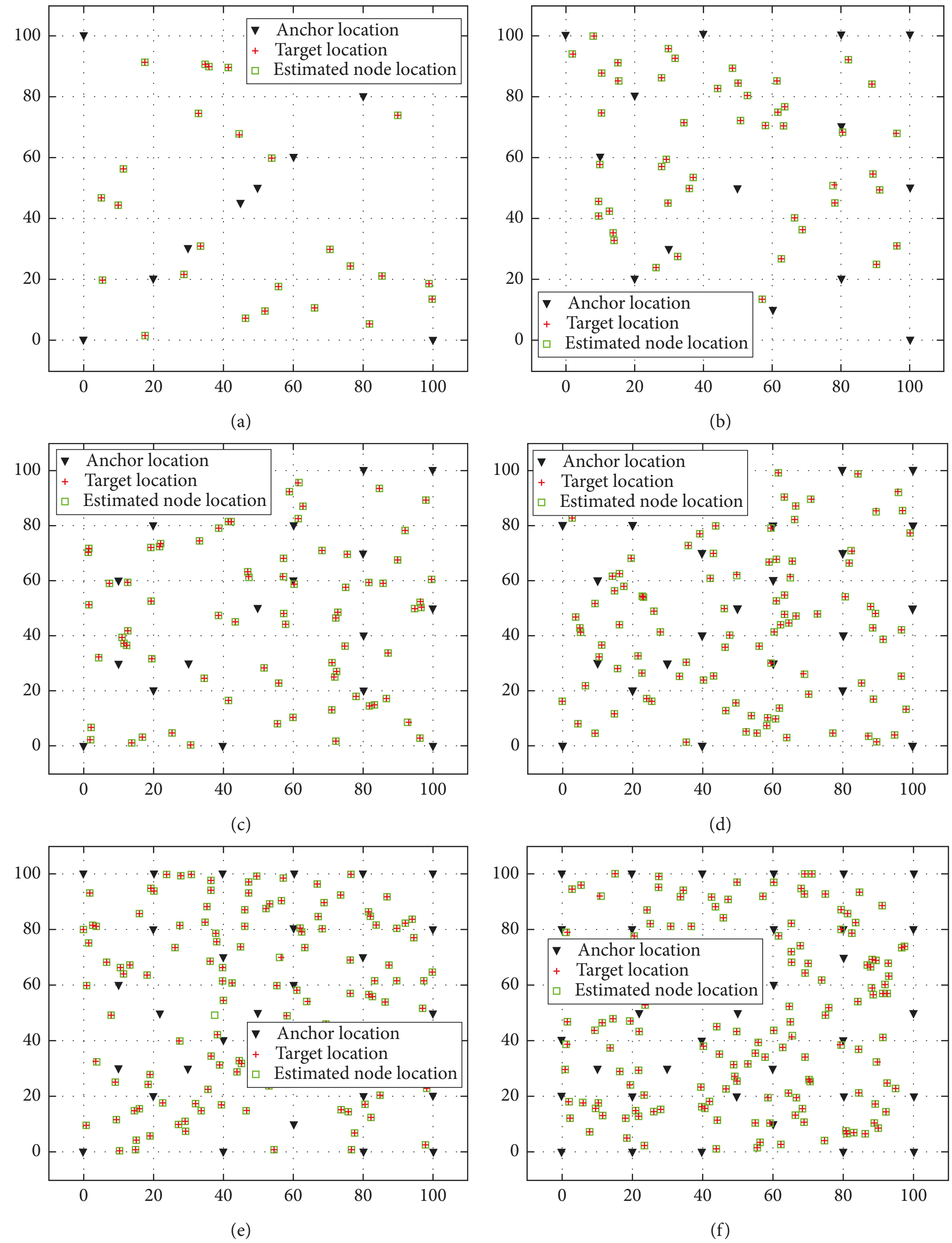

Figure 3: Node localization using different numbers of target nodes and anchor nodes. (a) Target nodes = 25; anchor nodes = 10. (b) Target nodes = 50; anchor nodes = 15. (c) Target nodes = 75; anchor nodes = 20. (d) Target nodes = 100; anchor nodes = 25. (e) Target nodes = 125; anchor nodes = 30. (f) Target nodes = 150; anchor nodes = 35.

increasing the number of iterations means higher amounts of computations, which requires longer computation time. On the other side, increasing the number of iterations means that the chance to find a better solution get bigger; hence, the number of localized nodes get larger and the value of localization error get smaller. For better results analysis, the experimental results are summarized in Table 3.

Based on Table 3, regarding the mean localization error (E_L), there is no clear pattern that can be detected to represent the relationship between this performance metric and the number of target nodes and anchor nodes. However, it is noticed that SSA has the best results regarding this performance metric compared to PSO, BOA, FA, and GWO, particularly, when the numbers of target nodes and anchor

Table 2: Performance metrics of different localization algorithms.

Target nodes	Anchor nodes	No. of iterations	PSO			BOA			FA			GWO			SSA		
			E_L(m)	T(s)	N_L	E_L(m)	T(s)	N_L	E_L(m)	T(s)	N_L	E_L(m)	T(s)	N_L	E_L(m)	T(s)	N_L
25	10	25	0.818	0.40	16	0.232	0.37	22	0.265	0.6	19	0.744	0.22	20	0.465	0.35	22
		50	0.812	0.40	15	0.225	0.38	23	0.262	1.2	20	0.741	0.41	21	0.462	0.35	23
		75	0.803	0.41	18	0.223	0.39	25	0.258	1.5	21	0.741	0.54	21	0.458	0.36	23
		100	0.792	0.41	18	0.221	0.40	25	0.251	1.8	19	0.740	0.79	23	0.451	0.37	24
50	15	25	0.419	0.71	41	0.338	0.84	47	0.477	1.6	46	0.690	0.42	44	0.477	0.67	43
		50	0.426	0.73	47	0.332	0.85	48	0.473	1.9	49	0.688	0.63	45	0.472	0.69	47
		75	0.429	0.76	46	0.326	0.86	49	0.465	2.5	49	0.686	0.81	46	0.468	0.69	48
		100	0.434	0.76	48	0.323	0.86	49	0.465	3.5	48	0.682	0.98	48	0.464	0.70	50
75	20	25	0.735	1.31	73	0.257	1.49	66	0.519	2.9	71	0.641	0.72	72	0.519	0.90	69
		50	0.728	1.32	74	0.257	1.49	66	0.513	3.8	72	0.641	0.95	72	0.513	0.92	72
		75	0.728	1.33	74	0.255	1.50	70	0.504	4.7	73	0.638	1.3	73	0.504	0.95	73
		100	0.724	1.35	75	0.253	1.52	72	0.503	5.2	73	0.635	1.4	74	0.503	0.96	75
100	25	25	0.661	2.10	97	0.355	2.40	97	0.711	3.8	98	0.611	1.1	95	0.511	1.31	98
		50	0.658	2.16	97	0.355	2.44	99	0.709	4.2	98	0.606	1.5	97	0.509	1.33	98
		75	0.642	2.17	99	0.333	2.47	100	0.702	5.6	99	0.602	1.8	98	0.502	1.36	99
		100	0.641	2.20	100	0.331	2.50	100	0.704	6.3	98	0.602	2.1	98	0.504	1.37	100
125	30	25	0.754	4.87	120	0.549	3.84	122	0.829	2.7	122	0.589	1.5	122	0.529	1.67	123
		50	0.748	4.86	121	0.548	3.85	123	0.824	4.5	123	0.580	2.2	123	0.524	1.68	124
		75	0.750	4.89	122	0.534	3.86	124	0.822	5.9	125	0.580	2.8	123	0.522	1.70	125
		100	0.752	4.95	125	0.534	3.89	124	0.822	6.5	124	0.572	3.3	125	0.522	1.72	125
150	35	25	0.625	5.41	145	0.766	5.88	147	0.911	2.5	149	0.559	2.8	148	0.511	2.12	149
		50	0.622	5.42	146	0.765	5.61	148	0.909	4.2	150	0.547	3.6	149	0.509	2.14	149
		75	0.619	5.44	148	0.763	5.64	149	0.904	6.4	150	0.523	4.3	150	0.504	2.16	150
		100	0.616	5.45	150	0.763	5.69	149	0.904	7.2	149	0.523	4.8	150	0.504	2.18	150

Table 3: Summary of experimental results of the different localization algorithms.

Target nodes	Anchor nodes	PSO			BOA			FA			GWO			SSA		
		E_L(m)	T(s)	N_L	E_L(m)	T(s)	N_L	E_L(m)	T(s)	N_L	E_L(m)	T(s)	N_L	E_L(m)	T(s)	N_L
25	10	0.79	0.40	18	0.22	0.38	25	0.25	1.8	19	0.74	0.54	21	0.45	0.35	24
50	15	0.43	0.76	46	0.32	0.86	49	0.42	2.5	49	0.69	0.81	46	0.46	0.69	48
75	20	0.72	1.35	75	0.25	1.52	68	0.51	4.7	73	0.64	0.95	72	0.50	0.96	73
100	25	0.65	2.16	100	0.35	2.45	100	0.70	5.3	98	0.60	2.1	98	0.51	1.35	100
125	30	0.74	4.90	123	0.54	3.87	124	0.82	6.5	124	0.58	2.8	123	0.52	1.70	125
150	35	0.62	5.43	149	0.76	5.65	149	0.90	7.2	149	0.52	4.3	150	0.50	2.15	150

nodes are increased. Regarding the computing time, it is noticed that increasing the number of target nodes and anchor nodes increase the computing time for all localization algorithms. However, once again, SSA has the best computing time compared to PSO, BOA, FA, and GWO. Finally, regarding the number of localized nodes (N_L), it is noticed that SSA has the best results compared to other localization algorithms. In addition, it is noticed that SSA has inferior E_L when the percentage of (number of anchor nodes/number of target nodes) becomes larger such as (10/25) in the first case in Table 3. That is because the localization accuracy increases when the anchor density increases with respect to the number of the target nodes [25].

The graphical representations of the experimental results for the different performance metrics are shown in Figures 4–6.

6. Conclusion

Accurate node localization concerns many applications that adopt WSNs. In this paper, a node localization algorithm has been proposed based on a novel bioinspired algorithm called Salp Swarm Algorithm (SSA) which handled the node localization problem as an optimization problem. The proposed algorithm has been implemented and validated in different WSN deployments using different numbers of target nodes and anchor nodes. Moreover, the proposed algorithm has been evaluated and compared to four well-known optimization algorithms, namely PSO, BOA, FA, and GWO, in terms of localization accuracy, computing time, and several localized nodes. The obtained simulation results have proved the superiority of the proposed algorithm compared to the other localization algorithms regarding the different performance metrics. In the future work, the proposed approach

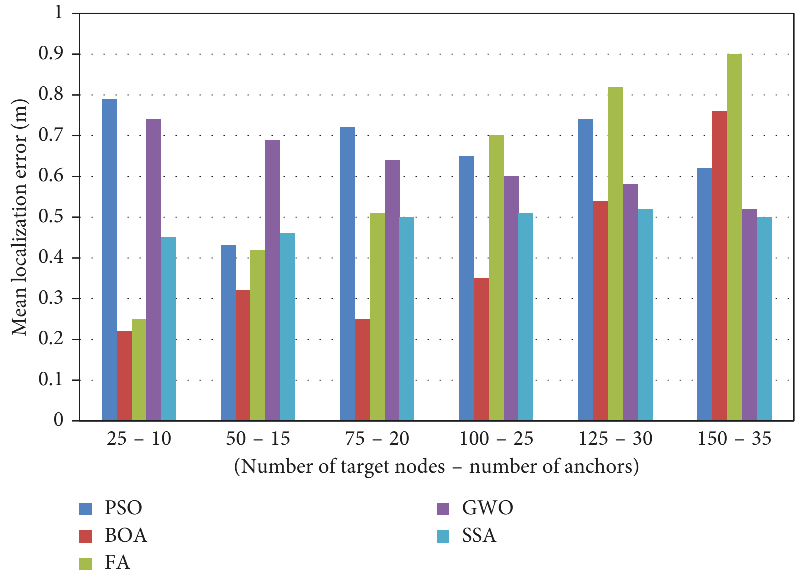

Figure 4: The localization error of the different localization algorithms in different WSN deployments.

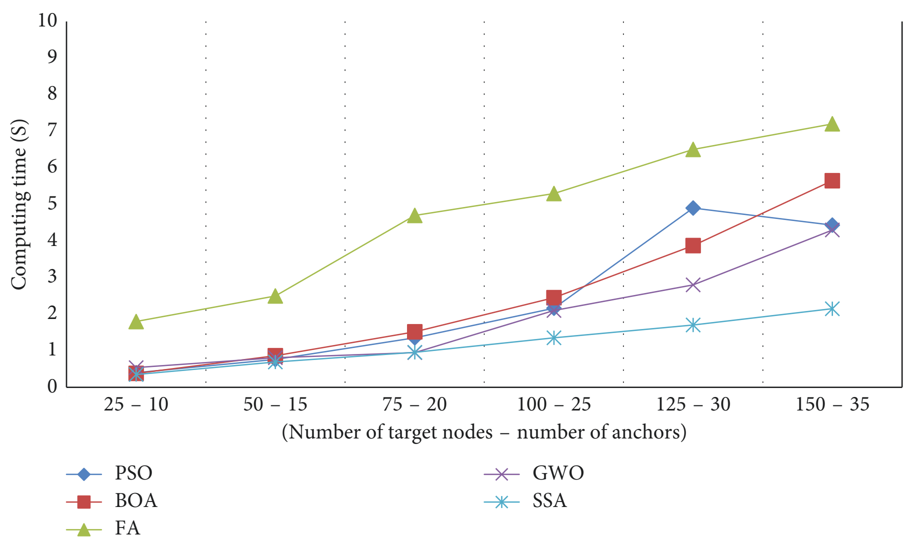

Figure 5: The computing time of the different localization algorithms in different WSN deployments.

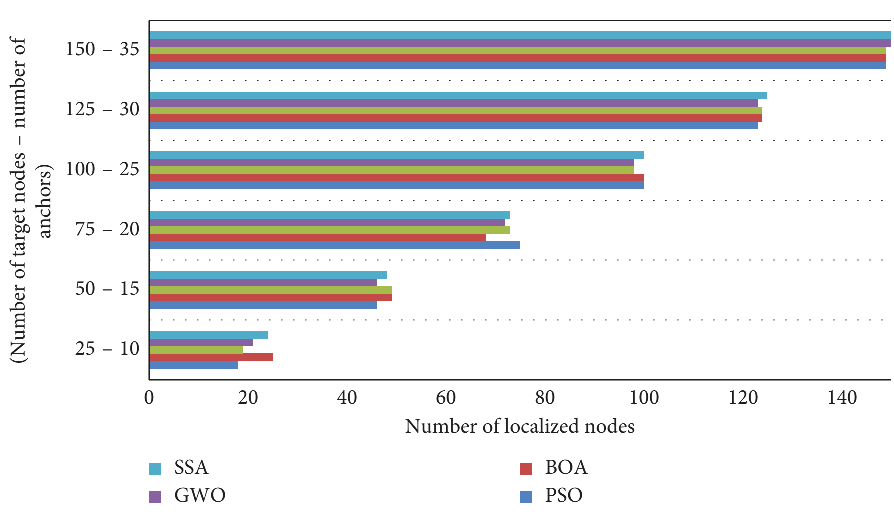

Figure 6: The number of localized nodes of the different localization algorithms in different WSN deployments.

can be hybridized with other algorithm to reduce the localization error.

Data Availability

No data were used to support this study. Because our article discusses the problem of localization in WSNs, our experimental results have been applied based on several networks' size.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. V. Kulkarni, G. K. Venayagamoorthy, and M. X. Cheng, "Bio-inspired node localization in wireless sensor networks," in *Proceedings of IEEE International Conference in Systems, Man, and Cybernetics (SMC)*, pp. 205–210, San Antonio, TX, USA, October 2009.

[2] D. Lavanya and S. K. Udgata, "Swarm intelligence-based localization in wireless sensor networks," in *International Workshop on Multi-Disciplinary Trends in Artificial Intelligence*, pp. 317–328, Springer, Berlin, Germany, 2011.

[3] J. Wang, R. K. Ghosh, and S. K. Das, "A survey on sensor localization," *Journal of Control Theory and Applications*, vol. 8, no. 1, pp. 2–11, 2010.

[4] J. Aspnes, T. Eren, D. K. Goldenberg et al., "A theory of network localization," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1663–1678, 2006.

[5] S. Goyal and M. S. Patterh, "Modified bat algorithm for localization of wireless sensor network," *Wireless Personal Communications*, vol. 86, no. 2, pp. 657–670, 2015.

[6] M. M. Ahmed, E. H. Houssein, A. E. Hassanien, A. Taha, and E. Hassanien, "Maximizing lifetime of wireless sensor networks based on whale optimization algorithm," in *International Conference on Advanced Intelligent Systems and Informatics*, pp. 724–733, Springer, Berlin, Germany, 2017.

[7] K. S. Low, H. A. Nguyen, and H. Guo, "A particle swarm optimization approach for the localization of a wireless sensor network," in *Proceedings of IEEE International Symposium in Industrial Electronics (ISIE)*, pp. 1820–1825, Vancouver, Canada, June 2008.

[8] D. Manjarres, J. Del Ser, S. Gil-Lopez, M. Vecchio, I. Landa-Torres, and R. Lopez-Valcarce, "On the application of a hybrid harmony search algorithm to node localization in anchor-based wireless sensor networks," in *Proceedings of IEEE International Conference in Intelligent Systems Design and Applications (ISDA)*, pp. 1014–1019, Córdoba, Spain, November 2011.

[9] M. Li, W. Xiong, and Q. Liang, "An improved abc-based node localization algorithm for wireless sensor network," in *Proceedings of IEEE International Conference in Wireless Communications, Networking and Mobile Computing (WiCOM)*, pp. 1–4, Barcelona, Spain, September 2012.

[10] A. Tamizharasi, R. Arthi, and K. Murugan, "Bio-inspired algorithm for optimizing the localization of wireless sensor networks," in *Proceedings of IEEE International Conference in Computing, Communications and Networking Technologies (ICCCNT)*, pp. 1–5, Tiruchengode, India, June 2013.

[11] C. Tang, R. Liu, and J. Ni, "A novel wireless sensor network localization approach: localization based on plant growth simulation algorithm," *Elektronika ir Elektrotechnika*, vol. 19, no. 8, pp. 97–100, 2013.

[12] O. D. Jegede and K. Ferens, "A genetic algorithm for node localization in wireless sensor networks," in *Proceedings of International Conference on Genetic and Evolutionary Methods (GEM), The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)*, pp. 1–7, Las Vegas, NV, USA, July 2013.

[13] S. Goyal and M. S. Patterh, "Wireless sensor network localization based on cuckoo search algorithm," *Wireless personal communications*, vol. 79, no. 1, pp. 223–234, 2014.

[14] L. Dan and W. Xian-bin, "An improved PSO algorithm for distributed localization in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 7, Article ID 970272, 2015.

[15] R. Krishnaprabha and A. Gopakumar, "Performance of gravitational search algorithm in wireless sensor network localization," in *Proceedings of IEEE National Conference in Communication, Signal Processing and Networking (NCCSN)*, pp. 1–6, Palakkad, India, October 2014.

[16] B. Peng and L. Li, "An improved localization algorithm based on genetic algorithm in wireless sensor networks," *Cognitive Neurodynamics*, vol. 9, no. 2, pp. 249–256, 2015.

[17] V. O. Sai, C. S. Shieh, T. T. Nguyen, Y. C. Lin, M. F. Horng, and Q. D. Le, "Parallel firefly algorithm for localization algorithm in wireless sensor network," in *Proceedings of IEEE International Conference in Robot, Vision and Signal Processing (RVSP)*, pp. 300–305, Kaohsiung, China, November 2015.

[18] S. Sivakumar and R. Venkatesan, "Meta-heuristic approaches for minimizing error in localization of wireless sensor networks," *Applied Soft Computing*, vol. 36, pp. 506–518, 2015.

[19] Z. Sun, L. Tao, X. Wang, and Z. Zhou, "Localization algorithm in wireless sensor networks based on multiobjective particle swarm optimization," *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, Article ID 716291, 2015.

[20] A. Arsic, M. Tuba, and M. Jordanski, "Fireworks algorithm applied to wireless sensor networks localization problem," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 4038–4044, Vancouver, Canada, July 2016.

[21] C. S. Shieh, V. O. Sai, Y. C. Lin, T. F. Lee, T. T. Nguyen, and Q. D. Le, "Improved node localization for WSN using heuristic optimization approaches," in *Proceedings of IEEE International Conference in Networking and Network Applications (NaNA)*, pp. 95–98, Hokkaido, Japan, July 2016.

[22] J. Cheng and L. Xia, "An effective Cuckoo search algorithm for node localization in wireless sensor network," *Sensors*, vol. 16, no. 9, p. 1390, 2016.

[23] P. T. Daely and S. Y. Shin, "Range based wireless node localization using Dragonfly Algorithm," in *Proceedings of IEEE International Conference in Ubiquitous and Future Networks (ICUFN)*, pp. 1012–1015, Vienna, Austria, July 2016.

[24] T. T. Nguyen, J. S. Pan, S. C. Chu, J. F. Roddick, and T. K. Dao, "Optimization localization in wireless sensor network based on multi-objective firefly algorithm," *Journal of Network Intelligence*, vol. 1, no. 4, pp. 130–138, 2016.

[25] S. Arora and S. Singh, "Node localization in wireless sensor networks using butterfly optimization algorithm," *Arabian Journal for Science and Engineering*, vol. 42, no. 8, pp. 3325–3335, 2017.

[26] S. R. Sujatha and M. Siddappa, "Node localization method for wireless sensor networks based on hybrid optimization of particle swarm optimization and differential evolution," *IOSR*

Journal of Computer Engineering, vol. 19, no. 2, pp. 07–12, 2017.

[27] R. Kaur and S. Arora, "nature inspired range based wireless sensor node localization algorithms," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, pp. 7–17, 2017.

[28] A. Tharwat, E. H. Houssein, M. M. Ahmed, A. E. Hassanien, and T. Gabel, "MOGOA algorithm for constrained and unconstrained multi-objective optimization problems," *Applied Intelligence*, vol. 48, no. 8, pp. 2268–2283, 2017.

[29] A. A. Ewees, M. Abd Elaziz, and E. H. Houssein, "Improved grasshopper optimization algorithm using opposition-based learning," *Expert Systems with Applications*, vol. 112, pp. 156–172, 2018.

[30] A. G. Hussien, E. H. Houssein, and A. E. Hassanien, "A binary whale optimization algorithm with hyperbolic tangent fitness function for feature selection," in *Proceedings of Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 166–172, Cairo, Egypt, December 2017.

[31] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: a multi-objective approach," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1656–1671, 2013.

[32] S. Arora and S. Singh, "Butterfly algorithm with levy flights for global optimization," in *Proceedings of IEEE International Conference in Signal Processing, Computing and Control (ISPCC)*, pp. 220–224, Waknaghat, India, September 2015.

[33] S. Arora and S. Singh, "An improved butterfly optimization algorithm with chaos," *Journal of Intelligent and Fuzzy Systems*, vol. 32, no. 1, pp. 1079–1088, 2017.

[34] R. B. Blair and A. E. Launer, "Butterfly diversity and human land use: species assemblages along an urban grandient," *Biological Conservation*, vol. 80, no. 1, pp. 113–125, 1997.

[35] X. S. Yang and X. He, "Firefly algorithm: recent advances and applications," 2013, https://arxiv.org/abs/1308.3898.

[36] X. S. Yang, "Firefly algorithms for multimodal optimization," in *Proceedings of International Symposium on Stochastic Algorithms*, pp. 169–178, Sapporo, Japan, October 2009.

[37] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[38] L. P. Madin, "Aspects of jet propulsion in Salps," *Canadian Journal of Zoology*, vol. 68, no. 4, pp. 765–777, 1990.

[39] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.

[40] P. A. Anderson and Q. Bone, "Communication between individuals in Salp chains. II. Physiology," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 210, no. 1181, pp. 559–574, 1980.

[41] H. T. Ibrahim, W. J. Mazher, O. N. Ucan, and O. Bayat, "Feature selection using salp swarm algorithm for real biomedical datasets," *IJCSNS*, vol. 17, no. 12, pp. 13–20, 2017.