



A comparative investigation of deterministic and metaheuristic algorithms for node localization in wireless sensor networks

Vaishali R. Kulkarni¹ · Veena Desai² · Raghavendra V. Kulkarni¹

Published online: 26 April 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Location-based services in wireless sensor networks demand precise information of locations of sensor nodes. Range-based localization, a problem formulated as a two-dimensional optimization problem, has been addressed in this paper as a multistage exercise using bio-inspired metaheuristics. A modified version of the shuffled frog leaping algorithm (MSFLA) has been developed for accurate sensor localization. The results of MSFLA have been compared with those of geometric trilateration, artificial bee colony and particle swarm optimization algorithms. Dependence of localization accuracies achieved by these algorithms on the environmental noise has been investigated. Simulation results show that MSFLA delivers the estimates of the locations over 30% more accurately than the geometric trilateration method does in noisy environments. However, they involve higher computational expenses. The MSFLA delivers the most accurate localization results; but, it requires the longest computational time.

Keywords Artificial bee colony algorithm · Particle swarm optimization algorithm · Sensor localization · Shuffled frog leaping algorithm · Trilateration · Wireless sensor networks

1 Introduction

Wireless sensor networks (WSNs) are ad hoc networks of massively deployed tiny, inexpensive and autonomous nodes used for cooperative monitoring of their environments. Each sensor node (a network node) can acquire sensory data, process it, and transmit the processed data to its peers over a wireless broadcast medium. WSNs have been successful in a broad spectrum of applications including surveillance, healthcare, environmental monitoring, structural health monitoring and home automation [43]. Since sensor nodes having extreme constraints on computational, memory and energy resources are deployed

for the monitoring of uncertain and harsh environments, any WSN operation must be performed in an energy-efficient manner. If not, nodes cease to function due to energy exhaustion thereby limiting the effectiveness of the WSN application.

In many WSN applications, nodes are deployed in a mission field randomly. The WSN topology emerges as the result of a distributed self-organization exercise. For successful self-organization, it is imperative that each node is aware of the precise coordinates of its location in the deployment field. The location information is useful in recognizing and correlating the data collected from sensors. It is also useful in location-aware data aggregation and routing protocols [2]. Moreover, the location itself is an important data queried by the user in several applications [36].

Localization refers to the post-deployment process of creating location awareness in as many sensor nodes as possible. Accurate localization significantly impacts the quality of detection and tracking applications of WSNs [29]. A straightforward solution to the localization problem is to equip each node with a global positioning system (GPS) hardware. This solution is costly, and results in

✉ Vaishali R. Kulkarni
vaishali@ieee.org

Veena Desai
veenadesai@git.edu

Raghavendra V. Kulkarni
arvie@ieee.org

¹ M S Ramaiah University of Applied Sciences, Bengaluru, India

² KLS Gogte Institute of Technology, Belagavi, India

increased bulk and energy expenses. It is also not a feasible solution if nodes are deployed in thick forests, caves or in closed indoor spaces from where they cannot access GPS satellites. In many localization techniques, a few special location-aware nodes called beacons or anchors are deployed in order to help the remaining ordinary nodes (also referred to as dumb, unsettled or unknown nodes) to create location awareness [8]. Achieving accurate, quick and energy efficient localization is a challenging task, and the same has been receiving considerable research attention.

Beacon-based localization can take place in a centralized or a distributed manner. In centralized approaches, a central node assists all the nodes in the WSN in localizing themselves. This method preempts the computation of the coordinates of the location at every sensor node. However, this method is computationally complex and has limited scalability. On the other hand, in distributed localization, each sensor node localizes itself independently using the information available locally. Distributed localization can be range-based or range-free. Range-based distributed localization is performed in two phases. In the first phase, each node estimates its distances from anchors in its vicinity. The received signal strength indication (RSSI), angle of arrival (AOA), time-difference-of-arrival (TDOA) and mapping techniques are used in distance estimation. In the second phase, the node estimates the coordinates of its location using the methods, such as triangulation, trilateration, multilateration or maximum likelihood estimation [8]. A node accomplishes this by solving a set of simultaneous equations, or through optimization algorithms that seek to minimize a suitably defined localization error. Generally, the localization of a large number of sensor nodes is achieved through iterative, multistage process in which a certain number of nodes get their location awareness with the help of beacons in the first stage. These settled nodes serve as beacons for the other unsettled nodes in the second stage. This process reiterates until either all nodes are localized, or no more nodes can be [22].

The problem of determining the x and y coordinates of the location of a sensor node in a flat mission field has been formalized traditionally as a two-dimensional optimization problem. The problem has been approached through several nature-inspired optimization algorithms. Algorithms from swarm intelligence (SI) paradigm of computational intelligence (CI) have been immensely popular because of their resource efficiency and the quality of solutions they offer [23]. Many SI-based localization algorithms have been introduced and compared amongst each other [26]. Two important questions arise here:

1. Is it possible to achieve more accurate localization using modified versions of the existing SI algorithms?

2. How do SI-based metaheuristic algorithms compare with the basic geometry-based deterministic localization algorithm?

These questions have been addressed in this article. Beacon-assisted, distributed, multistage localization using geometry-based deterministic and SI-based metaheuristic algorithms is the central theme of this article. Multistage localization has been performed first using the geometrical method. Further, a modified version of the shuffled frog leaping algorithm (SFLA), a metaheuristic inspired by social foraging of frogs, has been proposed for improved localization accuracy. The results of these two approaches have been compared with those of localization approaches proposed in previous literature, namely the particle swarm optimization (PSO) [21] and the artificial bee colony (ABC) algorithms [18]. The primary contributions of this article are:

1. The geometric trilateration method has been applied for the estimation of the coordinates of the locations of sensor nodes in WSNs.
2. A modified SFLA (MSFLA) has been proposed for the estimation of the coordinates of the locations of sensor nodes in WSNs.
3. Results of the localization methods based on geometric trilateration, PSO, ABC and MSFLA algorithms have been presented for WSNs having multiple node populations.
4. A comparative performance analysis of the geometry-based and CI-based approaches has been presented in terms of the number of localized nodes, localization accuracy and computing time.
5. An analysis of iterative propagation of localization error has been presented.
6. Scalability of the localization methods and the accuracy-speed tradeoff issue have been discussed.

The remainder of this article has been organized as follows: A brief survey of the previous research on sensor localization using deterministic and CI-based metaheuristic algorithms has been presented in Sect. 2. A mathematical overview of the multistage localization problem has been presented in Sect. 3. The geometrical trilateration method for sensor node localization has been discussed in Sect. 4. Brief introductions to the ABC and PSO algorithms and the detailed explanation of the proposed MSFLA for localization have been presented in Sect. 5. Details of the MATLAB-based numerical simulations have been presented and the results obtained have been compared in Sect. 6. Finally, concluding remarks have been presented and the directions for the future extension of this research have been suggested in Sect. 7.

2 Related work

Accurate estimation of locations of nodes plays a critical role in routing protocols, event detection, target tracking and many other functionalities of WSNs. Therefore, sensor localization has received a keen research attention since the early emergence of WSN deployments. Varieties of techniques and algorithms have been proposed by researchers working in the field of WSNs. Comprehensive taxonomies, surveys and comparative evaluations of these techniques have been appearing in literature periodically [1, 8, 15, 29, 36]. A few prominent, conventional, range-based and range-free localization algorithms are outlined here followed by some recent applications of CI algorithms in sensor localization.

Range-based localization involves the estimation of distances between beacons and dumb nodes. This has been accomplished through the methods, such as the measurement of received signal strength (RSS), AOA, TOA or TDOA. The estimates of distances are used to approximate the locations of sensors using geometric methods. These techniques are simple and efficient; but, they are very sensitive to the environmental noise. Range-based localization using RSSI, AOA [32], TOA [44] and TDOA [40] have been quite successful.

The effective localization algorithm presented in [31] extends GPS capabilities to non-GPS nodes in a WSN. In this approach, anchors flood the coordinates of their locations to all nodes in the WSN; then, each node estimates its location by triangulation to three anchors. This method is refined in [39], in which nodes improve their localization accuracy by measuring their distances from their neighbors. This results in the accumulation of errors. Error accumulation is tackled in [40] through Kalman filter-based least squares estimation. The problem of node localization is posed as a convex optimization based on semi-definite programming in [11].

Indoor localization is a research topic of special interest. This has been accomplished using the RSSI technique. This technique involves the use of the strength of a transmitted signal to estimate the distance between a transmitting node and a receiving node. The signal strength can be translated into the distance using a known radio propagation model. RADAR, a two-phase RSSI-based localization system used for indoor localization has been proposed in [4]. This approach has two phases: Calibration and localization. In the calibration phase, measurements of RSS at all positions in an entire building are stored in a database. In the localization phase, the node's location is determined as the best match with the database records. However, the accuracy of estimation is poor because the signal strength is influenced by the noise and the uncertainties of the propagation medium.

The limitation of the RSSI-based approach has been overcome by a distributed probabilistic approach in [37]. In this approach, localization has three phases: Calibration and static processing of RSS, localization with positive constraints and localization with negative constraints. Here, the measured distance is stored with positive and negative constraints. When the node has not been able to receive any RSS, then that signal is treated as a negative constraint. The uncertainties of the locations of the unknown node have been reduced using this probabilistic approach.

The centroid method is another beacon-based, range-free localization approach in which each node estimates its location using the locations of anchors in its reception range. The centroid of the anchor locations is taken as the estimate of the locations of the dumb node [9].

Accuracy of localization has been the major issue addressed in the previous research. The beacons that lie in a straight line, are referred as collinear beacons. If a node is assisted by three collinear beacons, two mirror reflected points across the line satisfy the distance constraints. This flip ambiguity results in large localization errors. A probabilistic notion of robust quadrilaterals has been presented in [30] to avoid large errors due to flip ambiguity. The unknown sensor nodes estimate the distances from their neighbors rather than beacons. The WSN is represented as a graph having approximate edge lengths. Each node computes the Euclidean position of each vertex. The localization algorithm presented in [30] estimates the positions accurately or not at all by refusing the nodes which have ambiguous positions. The need for beacons has been eliminated which makes it a low-cost algorithm. Further, the algorithm uses cluster-based localization to support the localization of mobile nodes.

Range-free localization techniques obviate the deployment of special beacon nodes. In the beacon-free localization presented in [38], nodes interact with their neighbors initially with the assumption of random coordinates. After initialization, a mass spring-based relaxation method is used to correct and balance the localization errors. Weighted centroid methods, distance vector (DV)-Hop and DV-Euclidian are some popular range-free localization techniques.

Since localization has been posed as a multidimensional optimization problem, SI-based optimization algorithms have been extensively used to address it. These algorithms are based on the collective behavior of populations of simple agents [25]. PSO [24], bacterial foraging algorithm [22] and ABC algorithm [26] have been successfully applied for sensor localization. The localization accuracies of these algorithms have been in the exact order in which the algorithms are listed above. However, the computational complexities of these algorithms for localization of a

fixed number of WSN nodes have been in the exact opposite order. The quest for higher accuracy and speed is driving researchers to develop variants and hybrids of these algorithms for WSN localization. The research reported in this article is an effort in this direction.

The SFLA is a recent algorithm developed for discrete and continuous optimization. A modified SFLA has been exploited to improve the accuracy of location of a sensor node using distance vector (DV) method in WSNs [13]. SFLA has shown better performance in comparison with least square method and the PSO algorithm in several applications. A quantitative cause-effect graph strategy and SFLA have been used to deploy new sensor nodes in [14]. A binary SFLA (BSFLA) has been developed with optimization of binary encoded problems in [6]. The BSFLA has been tested on a unit commitment problem to solve the operation and planning of a power system. The BSFLA has resulted in optimum results for this system.

The ABC algorithm is a simple and robust algorithm which has been applied in several optimization problems [5]. ABC has been successfully applied in image processing [28], bioinformatics [42] and engineering design [3]. ABC has been extensively hybridized with algorithms from other paradigms of CI, such as neural networks [16], fuzzy logic [19] and evolutionary algorithms [45]. Hybridization contributes to exploiting and exploring better solutions in a given search space. The ABC algorithm has been applied in optimal sensor deployment and accurate localization in [35]. In addition, the ABC algorithm has been used for optimal sensor deployment in irregular terrain by modelling the deployment challenge as a data clustering problem [41].

PSO is another simple SI-based algorithm that has been popular for its quick convergence and high-quality results. It has been used in minimization of localization error in WSNs [23]. A localization scheme using PSO and a bounding box approach has been presented in [27]. In the original bounding box method, the location is estimated using the intersection of all rectangular estimation values but in the approach presented in [27], the location is determined randomly within a rectangular area and then optimized by the PSO algorithm. The use of PSO has resulted in the enhancement of the accuracy and reduction of the computational efforts in comparison with the traditional bounding box approach. Node localization is addressed using a hybrid algorithm of PSO and Quasi-Newton algorithm in [10]. Hybridisation with PSO has resulted in fairly accurate localization and an improvement in speed.

In summary, CI-based optimization algorithms and their hybrids have been immensely popular in sensor localization. However, there is a constant quest for higher accuracy and lower computational cost. The MSFLA proposed in

this article is another step forward in the direction of enhancing localization accuracy.

3 Multistage localization

The mathematical details of the deployment scenario, the various steps involved in the multistage localization exercise, and the metrics of performance evaluation are presented in this section.

Deployment Scenario

The WSN deployment consists of U unknown (dumb) nodes and B beacons. The unknown nodes do not have the information about their locations. They are named as u_1, u_2, \dots, u_U .

Location-aware beacons are named as b_1, b_2, \dots, b_B . They are either deployed at known locations or they are equipped with GPS hardware. The communication range of each beacon is R meters; an unknown node farther than R meters from a beacon cannot hear the signals from the particular beacon. The objective of the localization exercise is to obtain accurate estimates of the x and y coordinates of the locations (u_{ix}, u_{iy}) of the i th node u_i , $i = 1, 2, \dots, U$. To achieve this, each unknown node u_i needs the information of the coordinates of at least three non-collinear beacons, (b_{jx}, b_{jy}) , $j = 1, 2, 3$.

Beacons and unknown nodes

Beacons broadcast coordinates of their locations periodically. An unknown node having three or more non-collinear beacons in its range is considered as a localizable node. If an unknown node is within the communication ranges of more than three non-collinear beacons, then the nearest three beacons are selected for localization.

Distance estimation

A dumb node estimates its distance \hat{d}_j from beacons using a measure, such as RSSI, ToA and TDoA. The distance measurement is not accurate due to the environmental noise power. Therefore, the distances are estimated in the simulation as $\hat{d}_j = d_j + g_j$, where d_j is the Euclidean distance obtained using (1) and g_j is the additive noise.

$$d_j = \sqrt{(b_{ix} - u_{jx})^2 + (b_{iy} - u_{jy})^2} + g_j \tag{1}$$

The value of g_j is generated through sampling the Gaussian distribution expressed in (2).

$$g_j \leftarrow \mathcal{N}(\mu, \sigma^2) \tag{2}$$

Here, \mathcal{N} represents the normal distribution, μ represents the mean, set to zero in this study, and σ^2 is the variance that represents the environmental uncertainties. The larger the variance, the more severe is the environmental noise. Thus, g_j accounts for the inaccuracy in distance estimation due to the environmental noise.

Location estimation

The estimated location of a dumb node u_i is $(\hat{u}_{ix}, \hat{u}_{iy})$, $i = 1, 2, \dots, U$. A node uses either the geometric method or one of the aforementioned SI algorithms to obtain $(\hat{u}_{ix}, \hat{u}_{iy})$. The metaheuristic methods seek to determine $(\hat{u}_{ix}, \hat{u}_{iy})$ such that the mean localization error E_i as expressed in (3) is minimum.

$$E_i = \frac{1}{3} \sum_{l=1}^3 \left(\sqrt{(\hat{u}_{ix} - b_{lx})^2 + (\hat{u}_{iy} - b_{ly})^2} - \hat{d}_i \right)^2 \tag{3}$$

Multistage iterative operation

Localization algorithms

Performance evaluation

The localization algorithm progresses in stages s_1, s_2, \dots, s_T . In the first stage s_1 , L_1 number of unknown nodes estimate their locations with the help of beacons. These nodes act as pseudo-beacons in the second stage s_2 . Thus, the number of anchors in s_2 is $B + L_1$. Due to the increased number of pseudo-beacons, more unknown nodes may get localized in s_2 . Thus, the number of beacons in stage s_3 rises to $B + L_1 + L_2$. This iterative process is terminated in stage s_T when all nodes are localized, or no more nodes can be. Performances of geometry-based localization and three metaheuristic algorithms, namely MSFLA, ABC and PSO are investigated. Each localizable node employs these techniques. The coordinates of beacons and the estimates of their distances are the inputs to these methods. The estimates of the coordinates $(\hat{u}_{ix}, \hat{u}_{iy})$ is the output.

At the end of the final stage of localization s_T , a total of L nodes are localized. The sum of squares of errors in the estimation, E , expressed in (4), is a metric of performance.

$$E = \frac{1}{L} \sum_{i=1}^L \sqrt{(u_{ix} - \hat{u}_{ix})^2 + (u_{iy} - \hat{u}_{iy})^2} \tag{4}$$

For an ideal localization exercise, $E = 0$. However, due to uncertainties in the environment, the method that results in the lowest value of E is considered the most accurate. In addition, the time T taken for the entire localization exercise is taken as a metric performance. The lower the value of T , the faster is the localization.

The pseudocode for stage-wise localization is depicted in Algorithm 1.

Algorithm 1 Localization

```

1: Initialize random beacon locations  $(b_{jx}, b_{jy})$ ,  $j = 1, 2, \dots, B$ ;
2: Initialize random locations for unknown nodes  $(u_{jx}, u_{jy})$ ,  $j = 1, 2, \dots, U$ ;
3: Initialize communication range ( $R$ ) and noise variance  $\sigma$ ;
4: Initialize maximum number of allowable iterations ( $k_{\max}$ );
5: Let  $L$  be the number of localizable nodes;
6: Let  $k = 1, L = 0, C_i = 0, i = 1, 2, \dots, U$ ;
7: while  $L \leq U$  OR  $k \leq k_{\max}$  do
8:   for  $i = i \leq U$  do
9:     for  $j = j \leq B$  do
10:      Estimate  $d_i$  using (1) and (2);
11:      if  $d \leq R$  then
12:         $C_i = C_i + 1$ ;
13:      end if
14:    end for
15:  end for
16:  for  $i = i \leq U$  do
17:    if  $C_i \geq 3$  then
18:       $L = L + 1$ ;
19:      Select the three nearest beacons;
20:      Compute  $(\hat{u}_{ix}, \hat{u}_{iy})$  using geometric method with the help of the three beacon locations;
21:      Use modified SFLA, ABC and PSO algorithms with input  $(b_{jx}, b_{jy})$ ,  $j = 1, 2, 3$  and  $d$ ;
22:      Determine  $(\hat{u}_{ix}, \hat{u}_{iy})$  that optimizes  $E$  using (3);
23:    end if
24:  end for
25:  Append  $(\hat{u}_{ix}, \hat{u}_{iy})$  to  $(b_{jx}, b_{jy})$ ;
26:   $k = k + 1$ ;
27: end while
28: Compute the mean square error of the entire localization exercise  $E$  using (4).
    
```

4 Sensor localization using geometric trilateration

In the geometric trilateration approach, every dumb node represents each anchor in its hearing range as a circle. The location of the anchor is the center of the circle, and the estimated distance between the dumb node and the anchor is the radius. A minimum of three anchors located on non-collinear points are necessary to estimate the locations of the dumb node. The location is estimated as the point of intersection of the three circles. The two steps involved in the estimation are discussed below.

4.1 Step 1: determine the point of intersection of two circles

This step involves determining the coordinates of the point of intersection of two circles. The intersection of circles C_1 and C_2 having radii of r_1 and r_2 meters, respectively, is

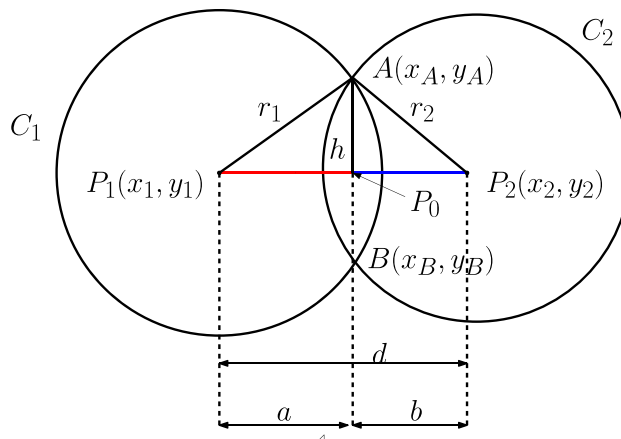


Fig. 1 Intersection of two circles

shown in Fig. 1. Let d be the distance between the circles' centers, $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$. Let d_x and d_y be the distances between the centers along x and y dimensions, determined as $d_x = (x_2 - x_1)$ and $d_y = (y_2 - y_1)$. From the Pythagorean theorem, $d = \sqrt{d_x^2 + d_y^2}$. The four cases in which the circles do not intersect with each other are:

1. If $d = (r_1 + r_2)$, the circles just touch each other;
2. If $d > (r_1 + r_2)$, the circles are disjoint;
3. If $d \leq |r_1 - r_2|$, a circle is contained fully within the other; and
4. If $d = 0$ and $r_1 = r_2$, then circles coincide fully.

If none of the above conditions is applicable, then the circles intersect each other at exactly two points, $A(x_A, y_A)$ and $B(x_B, y_B)$. The points A and B are symmetric with respect to the line joining the centers of the circles. Consider the triangles P_0P_1A and P_0P_2A shown in Fig. 1. Let $d = a + b$. The radii r_1 and r_2 , and the line segments a , b and h share the relations as expressed in (5) and (6).

$$r_1 = \sqrt{a^2 + h^2} \tag{5}$$

$$r_2 = \sqrt{b^2 + h^2} \tag{6}$$

Thus, a and b can be obtained using (7) and (8).

$$a = \frac{r_2^2 - r_1^2 + d^2}{2d} \tag{7}$$

$$b = \frac{r_1^2 - r_2^2 + d^2}{2d} \tag{8}$$

Let the line joining the points A and B intersect the line joining the centers of the circles at the point $P_0(x_0, y_0)$. The coordinates x_0 and y_0 of the point P_0 are obtained using (9) and (10).

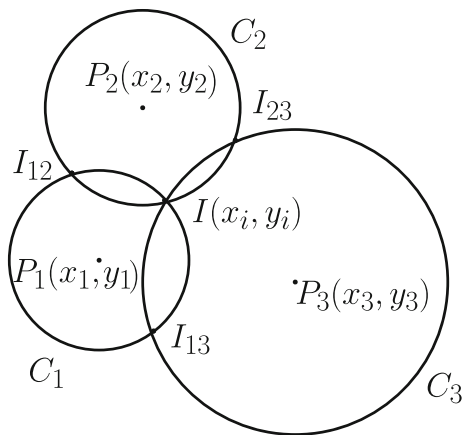


Fig. 2 Intersection of three circles

$$x_0 = x_1 + d_x \frac{a}{d} \tag{9}$$

$$y_0 = y_1 + d_y \frac{a}{d} \tag{10}$$

The coordinates of the point A are obtained using (11) and (12).

$$x_A = x_0 + d_x \frac{h}{d} \tag{11}$$

$$y_A = y_0 - d_y \frac{h}{d} \tag{12}$$

Similarly, the coordinates of the point B are obtained using (13) and (14).

$$x_B = x_0 - d_x \frac{h}{d} \tag{13}$$

$$y_B = y_0 + d_y \frac{h}{d} \tag{14}$$

4.2 Step 2: determine the point of intersection of three circles

This is a logical extension of Step 1. This involves determining the coordinates of the points of intersections of three circles taking two at a time. Three anchors are represented by circles C_1 , C_2 and C_3 centered at $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ and $P_3(x_3, y_3)$, respectively, as depicted in Fig. 2. Let the radii of these circles be r_1 , r_2 and r_3 meters, respectively. The coordinates of the six points of intersection of three pairs of circles (C_1, C_2), (C_1, C_3) and (C_2, C_3) are determined as discussed in Step 1. Since the three circles intersect at the point I , three out of the six intersection points obtained are identical. x_I and y_I , the coordinates of this point I , represent the location of the dumb node. This geometric trilateration method is deterministic, simple and straightforward. However, since the

estimation of the distances between anchors and dumb nodes involves large errors, the quality of localization obtained using this method is unlikely to be very accurate. That motivates the application of SI-based techniques for sensor localization in WSNs.

5 Sensor localization using ABC, PSO and MSFL algorithms

The SI-based algorithms used for multistage localization in this study are presented in the following subsections.

5.1 Sensor localization using the ABC algorithm

The ABC is a population-based algorithm that models the intelligent foraging behavior of honeybees. It has been used to address several multidimensional constrained optimization problems [20]. ABC comprises of three essential components: Food sources, employed foragers and unemployed foragers. Location of a food source represents a possible solution to the optimization problem. The fitness of the solution is related to the amount of nectar available in a food source. The Bees are referred as employed bees, onlookers and scouts and are used to bring the solution. Each employed bee modifies a position corresponding to the solution based on the visual information and evaluates the quantity of nectar amount. The position of new food source is generated using expression (15), where $k \in 1, 2, \dots, S$ and $j \in 1, 2, \dots, D$ are randomly chosen indexes. S and D represent the number of honeybees and the dimensionality of the search space, respectively.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{15}$$

The value of k is selected such that $k \neq i$ and $\phi \sim U[1, -1]$. Onlookers observe the food position brought by the employed bees, evaluate the nectar information taken from all employed bees and select the most feasible food source. If a solution representing a food source is not improved by a predetermined number of trials, then that food source is abandoned and the employed bee is converted to a scout bee. A control parameter referred as limit (L) is used for generating a scout bee. Scout bee produces new food position as expressed in (16), where $Q \sim U[1, -1]$.

$$x_i^j = x_{min}^j + Q(x_{max}^j - x_{min}^j) \tag{16}$$

The ABC algorithm fulfills the essential features of SI-based algorithms such as feedback about the solutions, fluctuation in solution by random values, sharing and multiple interactions [7]. Some of the drawbacks of ABC algorithms are: Population belonging to solution increases the computation cost. As compared with other SI-based

techniques, ABC requires more number of function evaluations and time. These disadvantages make ABC less attractive for applications which have complex objective functions.

5.2 Sensor localization using the PSO algorithm

The PSO algorithm includes a swarm of particles moving towards the areas of solution. Particles have been assigned with random positions and velocity at the initial stage. Each particle is evaluated through a fitness function. Each particle keeps track of its coordinates in the solution space which are associated with the best solution or the best fitness that has achieved so far. This value is referred as personal best (p_{id}). The best value obtained so far by any particle in the neighborhood is called global best g_d . With the change of position of each particle, the fitness is evaluated. The position of the particle is changed using velocity. Velocity V and position X are updated using (17) and (18) in each iteration. The update process is repeated until a fixed number of iterations k_{\max} have been reached or an acceptable g_d has been achieved.

$$V_{id}(k+1) = w \cdot V_{id}(k) + C_1 \cdot w_{1id}(k) \cdot (p_{id} - X_{id}) + C_2 \cdot w_{2id}(k) \cdot (g_d - X_{id}) \quad (17)$$

$$X_{id}(k+1) = X_{id}(k) + V_{id}(k+1) \quad (18)$$

Here, C_1 and C_2 are cognitive and social acceleration coefficients, respectively. The particles are accelerated towards p_{id} and g_d positions using these constants. $w_{1id}(k)$ and $w_{2id}(k) \sim U[0, 1]$. PSO has been effective in many optimization techniques to obtain optimal solutions. Many hybrid algorithms of PSO have been successfully used to solve optimization problems.

5.3 Sensor localization using the MSFLA

Bio-inspired algorithms can provide near-optimal solutions to data-driven problems having fluctuating constraints and incomplete information [17]. They are simple and resource-efficient. They have often provided higher quality of results than those of deterministic approaches to multi-dimensional optimization. The SFLA is a multidimensional discrete optimization algorithm that is gaining popularity in recent years [12]. The problem of sensor localization has been posed in this study as a 2-dimensional optimization problem. A modified SFLA has been used to determine the coordinates of dumb nodes such that the resulting localization error is minimum.

When an unknown node is in the range of three non-collinear beacons, it estimates its distance from beacons. This estimation is erroneous due to environmental factors which can be modelled with additive gaussian noise. The

known locations of beacons and the estimated distance from each beacon are given as the input parameters to the MSFLA. The algorithm aims at optimizing the localization error.

The original SFLA is a metaheuristic algorithm introduced in [12]. It combines the advantages of the PSO algorithm and the genetics-based memetic algorithm. The SFLA uses a virtual population of frogs that interact with each other and with their environment to search for the location that has an abundance of food. There is no centralized control that dictates the frogs' behavior. The interaction among the frogs leads to the emergence of intelligent global behavior. The food represents the fitness of a frog, which in turn refers to the location of a dumb sensor node in this study.

The population of frogs is divided into multiple subsets called memplexes. The frogs search for the food location within each memplex. A frog leaps from a memplex to another with a certain step size. This represents local search. During the search, the behavior of each frog gets influenced by that of the others in its memplex. This leads to the process of evolution. After a certain number of evolution steps, new memplexes are formed by shuffling the old ones. The processes of local search in each memplex and the shuffling of memplexes are repeated until the desired food quality is found or a predefined maximum number of evolution steps have been taken.

The mathematical details of the various steps involved in the SFLA are presented below:

Initialization

A population of P frogs are randomly initialized in a D -dimensional search space. The location of a frog i in the dimension d in the search space is represented as X_{id} , where $i = 1, 2, \dots, P$ and $d = 1, 2, \dots, D$. The search space is bound by the range $[X_{\min}, X_{\max}]$. The number of memplexes M and the maximum number of iterations allowed is k_{\max} . The step size used for a frog's leaping in search of a better food location is denoted by S . Each frog evaluates its fitness f using an objective function. X_l denotes the best fitness in one memplex. X_w represents the worst fitness in a memplex. The global best in

Local search and evolution

all the memplexes is denoted by X_g . In every iteration, fitnesses of X_l , X_g and X_w are determined as f_l , f_g and f_w , respectively. Initially, f_w is compared with f_l . If f_l is better than f_w , then a new S and new X_w is computed using (19). The new position of X_w is denoted by X_n . If f_w has not improved with the new X_n , then a new S is computed with X_g using (19).

$$S = \begin{cases} p_1(X_l - X_w) & \text{if } f_w < f_l \\ p_2(X_g - X_w) & \text{if } f_l < f_g \end{cases} \tag{19}$$

Here, $p_1, p_2 \sim U(0, 1)$, and $-S_{\max} \leq S_i \leq S_{\max}$. Frogs leap from a memplex to another using S_i in local search for food. Based on S_i , the better frog position X_n is computed using (20).

$$X_n = X_i + S_i \tag{20}$$

The fitness of the new frog position is denoted by f_n . If f_n is inferior in comparison with f_l and f_g , then frog positions are changed to random positions using (21).

$$X_n = p_3(X_{\max} - X_{\min}) + X_{\max} \tag{21}$$

Here, $p_3 \sim U(0, 1)$. The pseudocode of the SFLA has been presented in Algorithm 2.

Algorithm 2 SFLA

- 1: Initialize the frogs' positions X_i ;
- 2: Evaluate the fitnesses $f(\cdot)$ of the individual frogs in the population;
- 3: Sort the population in the decreasing order of their fitnesses;
- 4: **while** Convergence is not reached **do**
- 5: Divide P into M memplexes;
- 6: **for** Each $i=1$ to k_{\max} **do**
- 7: Determine local best, worst and global best frog positions, X_l , X_w and X_g , respectively;
- 8: Compute the step size using (19);
- 9: Compute X_n using (20);
- 10: **if** No improvement **then**
- 11: Discard old X_w and replace with new positions using (21);
- 12: **end if**
- 13: Replace X_i in their original locations in M_i and Sort M_i in decreasing order of fitness;
- 14: $i = i + 1$;
- 15: **end for**
- 16: Shuffle memplexes for global search;
- 17: Sort the population in the descending order of fitnesses and update X_l and X_g ;
- 18: **end while**

5.4 Proposed modification to the SFLA

The SFLA combines exploration and exploitation features of multidimensional optimization. It has been used for discrete and continuous optimization. This feature inspired the authors to modify SFLA to explore the local search in a more extensive manner. In the original SFLA, the new position for X_w is determined using X_l in its memplex. If there is no improvement in the fitness, then it is replaced by X_g . This is likely to leave better positions in the search space left unexplored resulting in premature convergence. A MSFLA has been proposed here to improve the exploring capacity of the frogs. This modification has been incorporated with an addition of a position vector \tilde{X}_i to hold the past history of X_l used for calculation of S . The S in the proposed MSFLA is determined using (22).

$$S_i = p_4(\tilde{X}_k - X_{wk}) \tag{22}$$

Here, $p_4 \sim U(0, 1)$. \tilde{X}_k corresponds to X_l , and X_{wk} represents X_w in the iteration k , $k = 1, 2, \dots, k_{\max}$. In each k , the X_l is added to the vector \tilde{X}_k . This modification enhances the local search and allows the use of SFLA for continuous optimization. The SFLA's drawback of premature convergence to local optima is thus overcome. The modification has shown improved accuracy. However the modification has resulted in a delayed convergence. Additional storage and computing time are necessary for the extra parameter \tilde{X}_k . Thus there is a trade-off between the speed and accuracy.

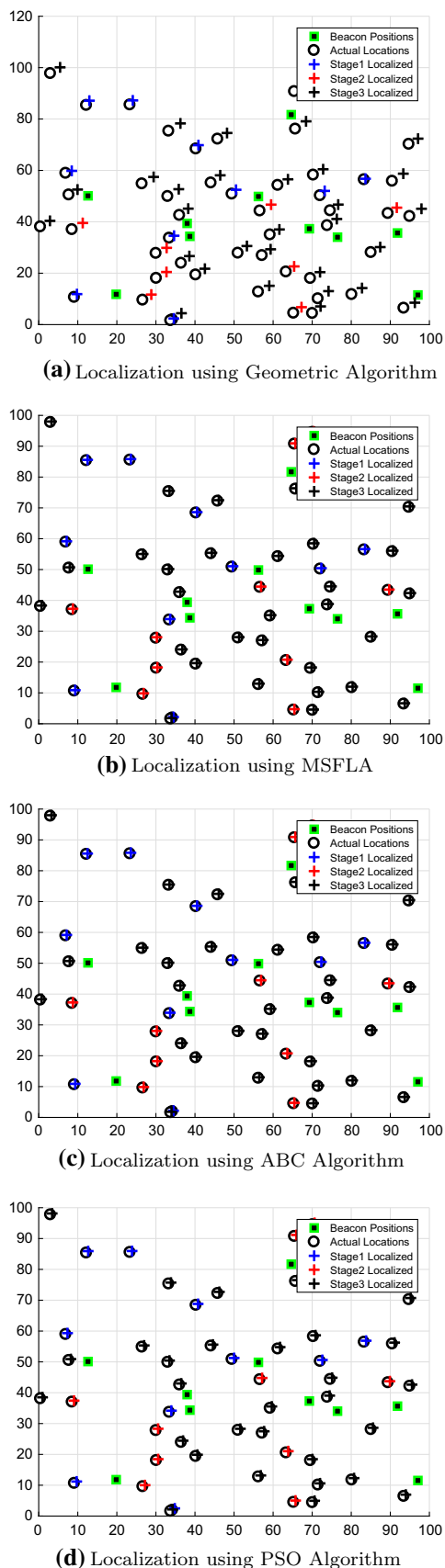


Fig. 3 Localization results for $U = 50$, $B = 10$, $R = 40$, $\sigma^2 = 2$

6 Numeric simulation and results

The details of numeric simulations and the results of multistage localization performed using the geometric and aforementioned metaheuristic methods have been presented in the following subsections.

6.1 Numeric simulation

All simulations have been carried out in MATLAB installed on a computer having on Intel® Core™ i5 processor @ 3.20 HZ and 8 gigabytes of RAM. A flat sensor deployment field of 100×100 square meters is considered. U dumb nodes and B beacons are deployed in the sensor field randomly. Experiments are carried out for multiple combinations of U and B . The accuracy of localization (E) and the computing time (T) are recorded as the performance metrics of localization. The discussion on localization using deterministic, MSFLA, ABC and PSO algorithms have been presented here.

6.1.1 Geometric trilateration-based localization

In this case study, each dumb node estimates its distances from all beacons in its hearing range. Then it chooses the three nearest beacons to localize itself. It localizes itself by substituting the coordinates of the beacons' locations and the estimated distances, as expressed in (3). When a stage of localization is complete, the localized nodes serve as beacons, and the process continues stage by stage until all localizable nodes estimate the coordinates of their locations.

6.1.2 Sensor localization using metaheuristic algorithms

In this case study, each dumb node uses a bio-inspired metaheuristic algorithm to localize itself in a distributed manner. The coordinates of the beacon locations and the estimated distances from beacons are the inputs to the algorithm. Solution candidates in each algorithm represent the coordinates of the location of the node. The objective function expressed in (3) is used as a measure of fitness of the solution candidates. The dynamics of the meta-heuristic algorithm propel the solution candidates iteratively to find the global minima point in the search space. This point represents the closest estimate of the location of the dumb node.

The computational complexity of the geometric trilateration algorithm is $O(n)$. Unlike metaheuristic algorithms, it is neither population-based, nor it requires additional space for storing the solutions of a previous iteration. However, in the case of metaheuristic algorithms, the

Table 1 Parameter setup for the deterministic, MSFLA, ABC and PSO algorithms in localization

Geometric	MSFLA	ABC	PSO
Communication radius, $R = 30, 40$ m	Number of memplexes, $m = 10$	Food sources, $N = 100$	Population of particles, $P = 100$
Noise variance σ^2 is varied from 1 to 10	Number of frogs in each memplex, $n = 10$	Population of employed bees and onlookers, $\frac{N}{2}$, Scout Bee, $s = 1$	Acceleration constants, $C_1 = C_2 = 2$
Dimensions, $D = 2$	Number of frogs, $f = m \times n$	Limit for abandonment, $l = ND$	Inertia weight, decreased linearly $w = 0.9$ to 0.4
Deployment area = 100×100 square meters	Number of iterations, $k_{\max} = 100$	Number of iterations, $(k_{\max}) = 100$	Number of iterations, $(k_{\max}) = 100$

Table 2 Results of multi-Stage localization for a sensor field with $X_{\min} = 0, X_{\max} = 100$ square meters, $U = 50, B = 10, R = 30$ m and $\sigma^2 = 0.2$

Trial	S_i	L_i	E_i				T_i			
			MSFLA	ABC	PSO	† Geo	MSFLA	ABC	PSO	Geo ^a
1	1	37	0.076	0.183	0.283	0.385	3.751	1.816	0.685	0.039
	2	50	0.052	0.099	0.173	0.855	7.669	5.514	2.551	0.099
2	1	42	0.099	0.838	0.21	0.926	4.598	2.127	0.815	0.032
	2	50	0.045	0.088	0.123	0.508	8.135	5.801	2.703	0.092
3	1	43	0.062	0.171	0.165	0.384	4.407	2.196	0.847	0.038
	2	50	0.059	0.054	0.139	0.441	8.019	5.912	2.810	0.096
4	1	46	0.057	0.099	0.246	0.400	4.551	2.129	0.873	0.040
	2	50	0.034	0.143	0.131	0.210	8.634	6.198	2.986	0.108
5	1	40	0.059	0.075	0.104	1.217	4.135	2.135	0.8432	0.038
	2	50	0.042	0.051	0.097	1.943	7.976	2.995	2.816	0.098

Numbers written in boldface digits signify the lowest value of the corresponding parameters
^aGeo geometric method

Table 3 A summary of results of 50 trial runs of SFL, ABC, PSO and geometric method for $D = 2$ and $x_{\min} = 0$ and $x_{\max} = 100$

	Localization error E				Computing time T			
	MSFLA	ABC	PSO	Geo ^a	MSFLA	ABC	PSO	Geo ^a
Mean	0.252	0.326	0.367	0.381	15.113	4.724	1.221	0.071
Std. Deviation	0.188	0.249	0.288	0.323	0.509	0.218	0.071	0.032

$U = 100, B = 30, \sigma^2 = 2, R = 30$

Numbers written in boldface digits signify the lowest value of the corresponding parameters
^aGeo geometric method

number of iterations is generally not fixed. Some metaheuristic algorithms need thousands of iterations involving significant computational effort in each iteration. Whereas, others may use millions of iterations involving a little effort for each iteration. Thus, the computational complexities of metaheuristic algorithms are generally compared empirically using computational time or the number of evaluations of the fitness function. The ABC and the PSO algorithms have lesser numbers of steps and storage variables in comparison with MSFLA. The MSFLA requires the storage of the solutions of the previous search operations to select better results. This results in the higher number of iterations and more memory requirement than in

the other algorithms [34]. The computational complexity of each metaheuristic algorithm is $O(d \times p + C \times p)$, where d is the dimensionality of the problem, p and C represent the population and cost of the fitness function, respectively [33].

The results of node localization using geometric trilateration, MSFLA, ABC and PSO algorithms are discussed in Sect. 6.2.

6.2 Results and discussion

Locations of beacons and the locations of dumb nodes estimated in each stage by deterministic, MSFLA, ABC

and PSO algorithms in a trial run are depicted in Fig. 3. The parameter setup for each algorithm used in localization is presented in Table 1. These parameters are chosen as suggested in the previous research literature. All the four methods have exhibited acceptable accuracy. However MSFLA is lot more accurate than the other three algorithms. A statistical summary of the results of localization performed using the four methods is presented in Table 3.

The comparison of deterministic and meta-heuristic algorithm has been presented in terms localization error (E) and computing time (T). Localization process is in multiple stages. The results have been recorded in each stage S_i . Terms E_i and T_i denote the computing time and the localization error in stage S_i , respectively. Lower E_i and T_i represents more accurate and quicker localization. Further the parameter L_i denotes the number of nodes that are localized in stage S_i . Higher L_i represents more number of localized nodes. The details of the L_i , E_i and T_i have been presented in Table 2.

The statistics of the results of the deterministic and bio-inspired meta-heuristic algorithm localization is depicted in Table 3. Table 3 points out the trade-off between the localization accuracy and computing time.

The performance of MSFLA is superior to those of the other localization algorithms in terms of localization accuracy. Thus MSFLA is recommended in WSN applications in which require accurate localization is critically important. However, MSFLA is slowest among the algorithms studied here. The computing time of all algorithms are recorded to study their speed of location. The following is the list of algorithms arranged in the ascending order of computing times.

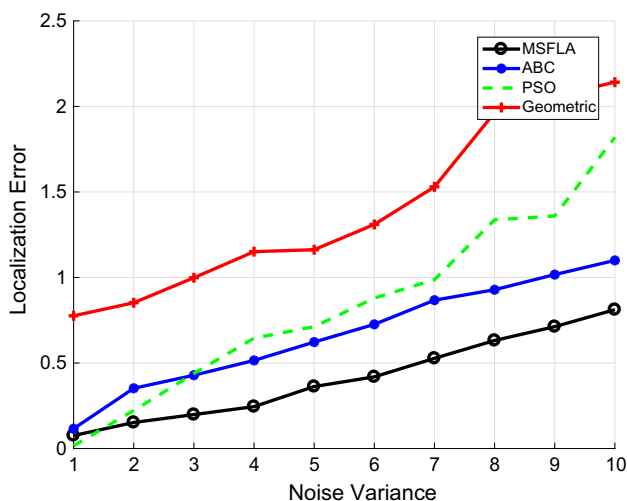


Fig. 4 Dependence of localization error on the environmental noise variance

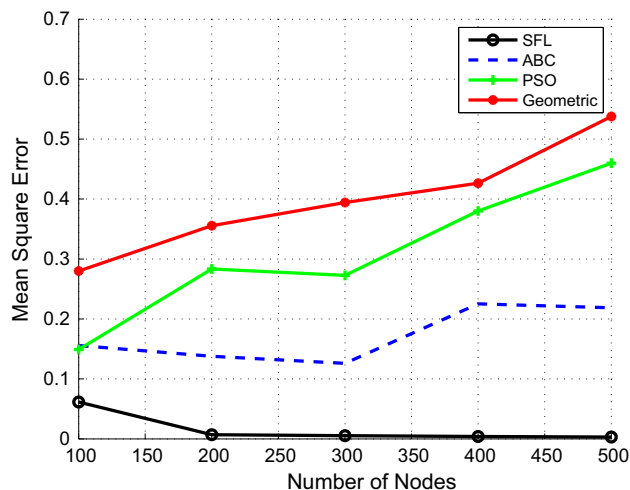


Fig. 5 Dependence of localization error with varying node density

1. Deterministic algorithm
2. The PSO algorithm
3. The ABC algorithm
4. The MSFLA

The geometric trilateration algorithm is the fastest of the algorithms considered in this study. The high speed of the geometrical method is attributed to the fact that it does not involve population-based iterative search. Therefore, it is most suitable in WSN applications that require quick localization.

The PSO algorithm is faster in convergence than the other metaheuristic algorithms. The ABC algorithm involves a step in which the entire population of candidate solutions is discarded and a new swarm is created if the optimal solution has not been obtained in a given number of iterations. This results in slower convergence. The MSFLA requires the storing of local best solutions of previous iterations. This adds to computational expenses in each iteration. This causes the MSFLA to be the slowest.

The variation of localization error E with increase in the environmental noise variance σ^2 has been depicted in Fig. 4. The localization error E increases with the increase in environmental noise variance σ^2 . The results depicted in Fig. 4 show that the localization using metaheuristic algorithms is more accurate than the analytic method.

Scalability plays an important role in the WSN applications. In order to study the scalability, multiple experiments have been conducted with increasing number of dumb nodes. The results have been depicted in Fig. 5. The performance of the MSFLA is consistently better than other algorithms in all the experiments in terms of accuracy of localization.

Heuristic algorithms result in the lower localization error E with the increase in the density of nodes. This is depicted in Table 4. With the geometric method, the value

Table 4 Statistical summary of localization accuracy and computing time with increasing density of unknown nodes and beacons in 50 trial runs

Parameter	E				T			
	MSFLA	ABC	PSO	Geo ^a	MSFLA	ABC	PSO	Geo ^a
Node density								
$U = 50, B = 10$	0.088	0.150	0.182	1.166	7.157	4.711	2.209	0.109
$U = 100, B = 20$	0.034	0.050	0.093	2.297	16.299	11.881	5.742	0.215
$U = 200, B = 40$	0.016	0.027	0.073	3.412	53.457	48.07	24.543	1.004
$U = 500, B = 100$	0.013	0.029	0.079	4.059	75.508	55.799	26.191	1.099
$U = 1000, B = 200$	0.006	0.026	0.069	10.11	221.007	193.631	97.403	3.860

^aGeo geometric method

of E increases when the network is scaled to a larger size. Metaheuristic algorithms result in improved accuracy with the increase in number of nodes. In the geometric trilateration method, the locations are estimated using geometrical information and does not involve any error optimization. Thus metaheuristic algorithms are more suitable than the conventional deterministic algorithms when the WSN has higher node population.

Multistage localization results in considerable saving in the cost of hardware for localization. The localization cost is determined as the cost of installation of GPS hardware for localization. The cost of localization is higher, if the GPS hardware is installed on each node. The multi-stage localization considered here requires GPS hardware only on beacons. Therefore the total cost of localization is considerably low.

7 Conclusion

The performances of geometry-based and bio-inspired metaheuristic algorithms have been compared in this paper for the localization of sensor nodes in WSNs. First, unknown nodes are localized using the geometrical trilateration method. Further, the localization problem has been formulated as a two-dimensional optimization problem and addressed using SI-based metaheuristic algorithms. A brief background of multistage localization has been presented. An modified SFLA has been proposed and the same is applied for sensor localization. The results of localization using MSFLA have been compared with that of localization using ABC and PSO algorithms. The MSFLA-based localization results very high accuracy localization. The other metaheuristic algorithms show better performance than the geometrical method in terms of accuracy. However, metaheuristic algorithms result in slower localization because of their population-based, iterative nature. Therefore, they are recommended in WSN deployments in which accuracy of localization is extremely important. On the other hand, geometrical trilateration is recommended for WSN deployments that require quick localization.

This study can be extended further in multiple directions. Localization of mobile sensor nodes is a natural extension of this investigation. The development of algorithms to determine optimal path for a mobile beacon is another direction in which this research can be extended. Further, quick and accurate algorithms can be developed by hybridizing SI algorithms with those from the other paradigms of CI for the sensor localization. Lastly, issues, such as security and energy expenditure in localization of sensor nodes in WSNs can be considered in the development of the objective functions used in metaheuristic algorithms.

Acknowledgements Authors gratefully acknowledge the support received from M. S. Ramaiah University of Applied Sciences, Bengaluru, India, and KLS Gogte Institute of Technology, Belagavi, India.

References

- Alrajeh, N. A., Bashir, M., & Shams, B. (2013). Localization techniques in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 9(6), 304628.
- Aspnes, J., Eren, T., Goldenberg, D. K., Morse, A. S., Whiteley, W., Yang, Y. R., et al. (2006). A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12), 1663–1678.
- Aydin, D. (2015). Composite artificial bee colony algorithms: From component-based analysis to high-performing algorithms. *Applied Soft Computing*, 32(C), 266–285.
- Bahl, P., & Padmanabhan, V. N. (2000). RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)* (pp. 775–784).
- Bansal, J. C., Sharma, H., & Jadon, S. S. (2013). Artificial bee colony algorithm: A survey. *International Journal of Advanced Intelligence Paradigms*, 5, 123–159.
- Barati, M., & Farsangi, M. M. (2014). Solving unit commitment problem by a binary shuffled frog leaping algorithm. *IET Generation, Transmission Distribution*, 8(6), 1050–1060.
- Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial systems*. New York, NY: Oxford University Press Inc.
- Boukerche, A., Oliveira, H. A. B. F., Nakamura, E. F., & Loureiro, A. A. F. (2007). Localization systems for wireless sensor networks. *IEEE Wireless Communications*, 14(6), 6–12.
- Bulusu, N., Heidemann, J., & Estrin, D. (2000). GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5), 28–34.

10. Cao, J. (2015). A localization algorithm based on particle swarm optimization and quasi-Newton algorithm for wireless sensor networks. *Journal of Communication and Computers*, 10(2), 85–90.
11. Doherty, L., Pister, K., & El Ghaoui, L. (2001). Convex position estimation in wireless sensor networks. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)* (Vol. 3, pp. 1655–1663).
12. Eusuff, M., Lansey, K., & Pasha, F. (2006). Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. *Engineering Optimization*, 38, 129–154.
13. Feng, C., & Zhang, L. H. (2012). A modified shuffled frog leaping algorithm for solving nodes position in wireless sensor network. *Proceedings of the International Conference on Machine Learning and Cybernetics*, 2, 555–559.
14. He, K., Jia, M., & Xu, Q. (2016). Optimal sensor deployment for manufacturing process monitoring based on quantitative cause-effect graph. *IEEE Transactions on Automation Science and Engineering*, 13(2), 963–975.
15. Hightower, J., & Borriello, G. (2001). Location systems for ubiquitous computing. *Computer*, 34(8), 57–66.
16. Hsieh, T. J., Hsiao, H. F., & Yeh, W. C. (2011). Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. *Applied Soft Computing*, 11(2), 2510–2525.
17. Kar, A. K. (2016). Bio inspired computing—A review of algorithms and scope of applications. *Expert Systems with Applications*, 59, 20–32.
18. Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), 459–471.
19. Karaboga, D., & Ozturk, C. (2010). Fuzzy clustering with artificial bee colony algorithm. *Scientific Research and Essays*, 5(14), 1899–1902.
20. Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2014). A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Revolution*, 42(1), 21–57.
21. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 4, 1942–1948.
22. Kulkarni, R. V., & Venayagamoorthy, G. K. (2010). Bio-inspired algorithms for autonomous deployment and localization of sensor nodes. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(6), 663–675.
23. Kulkarni, R. V., & Venayagamoorthy, G. K. (2011). Particle swarm optimization in wireless-sensor networks: A brief survey. *Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 41(2), 262–267.
24. Kulkarni, R. V., Venayagamoorthy, G. K., & Cheng, M. X. (2009). Bio-inspired node localization in wireless sensor networks. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics* (pp. 205–210).
25. Kulkarni, R. V., Förster, A., & Venayagamoorthy, G. K. (2011). Computational intelligence in wireless sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 13(1), 68–96.
26. Kulkarni, V. R., Desai, V., & Kulkarni, R. V. (2016). Multistage localization in wireless sensor networks using artificial bee colony algorithm. In *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–8).
27. Li, H., Xiong, S., Liu, Y., Kou, J., & Duan, P. (2011). A localization algorithm in wireless sensor networks based on PSO. In *Proceedings of the International Conference on Advances in Swarm Intelligence, Part II* (pp. 200–206). Berlin: Springer.
28. Ma, M., Liang, H., Jian, Guo M., Fan, Y., & Yin, Y. (2011). SAR image segmentation based on artificial bee colony algorithm. *Applied Soft Computing*, 11(8), 5205–5214.
29. Mao, G., & Fidan, B. (2009). *Localization algorithms and strategies for wireless sensor networks*. Hershey, PA: Information Science Reference - Imprint of: IGI Publishing.
30. Moore, D., Leonard, J., Rus, D., & Teller, S. (2004). Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys '04* (pp. 50–61).
31. Niculescu, D., & Nath, B. (2001). Ad hoc positioning system (APS). *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, 5, 2926–2931.
32. Niculescu, D., & Nath, B. (2003). Ad hoc positioning system (APS) using AOA. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)* (Vol. 3, pp. 1734–1743).
33. Oliveto, P. S., He, J., & Yao, X. (2007). Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results. *International Journal of Automation and Computing*, 4(3), 281–293.
34. Ong, Y.-S., Zhu, N., Lim, M.-H., & Wong, K. W. (2006). Classification of adaptive memetic algorithms: A comparative study. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(1), 141–152.
35. Öztürk, C., Karaboğa, D., & Gökemli, B. (2012). Artificial bee colony algorithm for dynamic deployment of wireless sensor networks. *Turkish Journal of Electrical Engineering & Computer Sciences*, 20(2), 255–262.
36. Patwari, N., Ash, J. N., Kyperountas, S., Hero, A. O., Moses, R. L., & Correal, N. S. (2005). Locating the nodes: Cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4), 54–69.
37. Peng, R., & Sichertiu, M. L. (2007). Probabilistic localization for outdoor wireless sensor networks. *SIGMOBILE Mobile Computation and Communication Review*, 11(1), 53–64.
38. Priyantha, N. B., Balakrishnan, H., Demaine, E., & Teller, S. (2003). Poster abstract: Anchor-free distributed localization in sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems* (pp. 340–341).
39. Savarese, C., Rabaey, J., & Langendoen, K. (2002). Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *Proceedings of the USENIX Technical Annual Conference* (pp. 317–327).
40. Savvides, A., Han, C. C., & Strivastava, M. B. (2001). Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, New York, NY, USA (pp. 166–179).
41. Udgata, S. K., Sabat, S. L., & Mini, S. (2009). Sensor deployment in irregular terrain using artificial bee colony algorithm. In *World Congress on Nature Biologically Inspired Computing, NaBIC 2009* (pp. 1309–1314).
42. Vargas Benítez, C. M., & Lopes, H. S. (2010). Parallel artificial bee colony algorithm approaches for protein structure prediction using the 3DHP-SC model. In *Intelligent Distributed Computing IV: Proceedings of the 4th International Symposium on Intelligent Distributed Computing*, Tangier, Morocco. Berlin: Springer (pp. 255–264).
43. Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12), 2292–2330.
44. Youssef, M., Youssef, A., Rieger, C., Shankar, U., & Agrawala, A. (2006). Pinpoint: An asynchronous time-based location determination system. In *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services (MobiSys)* (pp. 165–176). New York, NY: ACM.

45. Zhao, H., Pei, Z., Jiang, J., Guan, R., Wang, C., & Shi, X. (2010). A hybrid swarm intelligent method based on genetic algorithm and artificial bee colony. In *Proceedings of the First International Conference, Advances in Swarm Intelligence (ICSI)* (pp. 558–565). Berlin: Springer.



Vaishali R. Kulkarni received her B.E. degree in Computer Science and Engineering from Karnatak University, India, and M. Tech. degree in Computer Science and Engineering from Visvesvaraya Technological University, India. She is pursuing her Ph.D. degree in Computer Science and Engineering from the Visvesvaraya Technological University, India. Her research interests include wireless sensor networks, computational intelligence, swarm

intelligence, fuzzy logic, and neural networks. She is a member of IEEE, a life member of the Indian Society for Technical Education (ISTE) and a member of Computer Society of India (CSI). She has served as invited reviewer, session chair and author in international conferences and journals. She has published two refereed journal papers and ten refereed conference proceedings papers.



Veena Desai received her B.E. degree in Electronics and Communication Engineering from Karnatak University, India, and M.Tech. degree in Computer Networks Engineering from Visvesvaraya Technological University, India. She received her Ph.D. degree in Electronics and Communication Engineering from the Visvesvaraya Technological University, India. Her research interests include computer networks, wireless networks and cryptography. She

is a member of IEEE and Computer Society of India (CSI). She has published a book chapter, twelve refereed journal papers and twelve refereed conference proceedings papers.



Raghavendra V. Kulkarni received his B.E. degree in Electronics and Communication Engineering from Karnatak University, India, and M.Tech. degree in Electronics Engineering from the Institute of Technology, Banaras Hindu University, India. He received his Ph.D. degree in Electrical Engineering from the Missouri University of Science and Technology, Rolla, USA. His research interests include the development of wireless sensor

network applications using computational intelligence tools. He is a senior member of IEEE, a life member of the Indian Society for Technical Education (ISTE), a member of the IEEE Computational Intelligence Society (CSI), and a member of Computer Society of India (CSI). He has published a book chapter, over twelve refereed journal papers and sixteen refereed conference proceedings papers.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.