Arab J Sci Eng (2017) 42:3325–3335
DOI 10.1007/s13369-017-2471-9

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

Node Localization in Wireless Sensor Networks Using Butterfly Optimization Algorithm

Sankalap Arora[1] · **Satvir Singh**[2]

Received: 3 November 2016 / Accepted: 6 March 2017 / Published online: 18 March 2017

Abstract Accurate localization of sensor nodes has a strong influence on the performance of a wireless sensor network. In this paper, a node localization scheme using the application of nature-inspired metaheuristic algorithm, i.e., butterfly optimization algorithm, is proposed. In order to validate the proposed scheme, it is simulated on different sizes of sensor networks ranging from 25 to 150 nodes whose distance measurements are corrupted by gaussian noise. The performance of the proposed novel scheme is compared with performance of some well-known schemes such as particle swarm optimization (PSO) algorithm and firefly algorithm (FA). The simulation results indicate that the proposed scheme demonstrates more consistent and accurate location of nodes than the existing PSO- and FA-based node localization schemes.

Keywords Wireless sensor networks (WSN) · Node localization · Butterfly optimization algorithm (BOA) · Particle swarm optimization (PSO) algorithm · Firefly algorithm (FA)

1 Introduction

Wireless sensor network (WSN) is an emerging technology that has potential applications in various fields like healthcare, surveillance, astronomy, military, and agriculture [1]. WSN has wide application prospects due to its fast and easy deployment, and self-organization. WSN consists of a large number of tiny, inexpensive autonomous sensor nodes (either homogenous or heterogeneous) to observe physical and environmental conditions [2]. These autonomous nodes sense, process and pass the gathered data from the environment over wireless ad hoc network to the base station or the sink node which act as the final destination [3]. The different optical, biological, chemical, magnetic sensors can be appended to the nodes to compute environmental properties. The features of WSN like self-organization and rapid deployment makes it promising for most of the WSN applications. In the application of WSN, sensor nodes sense and report the events of interest which can be examined when the position of target nodes reporting the event is known. The estimation of the sensor nodes is one of the most important issues of the WSN and is known as localization problem [4].

The technology of node localization can locate and track nodes, so that the monitoring data are more meaningful, i.e., data gathered at sink node will be meaningless to the user without localization information of the nodes in the sensor field. The localization can be defined as determination of the position of the unknown sensor nodes called as target nodes using the known position of the sensor nodes called as anchor nodes based on the measurement such as time difference of arrival, time of arrival, angle of arrival, triangulation and maximal likelihood etc. [5]. The localization issue of WSN can be resolved by using global positioning system (GPS) with each sensor node, but this is not favored due to energy, cost and size issues. It even does not work properly indoor and underwater. So, efficient and better alternative is required to localize the sensor nodes. Various non-GPS-based localization algorithms can be used which is categorized into range-based and range-free algorithms [6]. Range-based localization algorithms use point-to-point distance estimation or angle-based estimation between sensor nodes. In this,

✉ Sankalap Arora
sankalap.arora@gmail.com

Satvir Singh
DrSatvir.in@gmail.com

1 I.K. GUJRAL Punjab Technical University, Jalandhar, Punjab, India

2 SBSSTC, Ferozpur, Punjab, India

location is estimated with the help of trilateration of anchor nodes (whose position is known). Range-free localization algorithms do not require range information between target node and anchor node, but depends on the topological information. Range-based algorithms provide more accuracy as compared to range-free localization algorithms, but they are not so economic [7].

A WSN consists of n nodes which are distributed in two-dimensional field. Each node has a communication range of r. The WSN can be represented as a Euclidean graph $G = (V, E)$, where V is the set of sensor nodes and $(i, j) \in E$, if the distance between ith sensor node and jth sensor node is r . Target nodes are the set T of nodes which do not know their position in the network, whereas localized nodes are the set L of nodes which estimated their position using some localization technique. Now, consider a WSN $G = (V, E)$, and a set of anchor nodes A and their positions (x_a, y_a), for all $a \in A$, it is desired to find the position (x_t, y_t) of as many $t \in T$ as possible, transforming the target nodes into localized nodes L [8].

The range-based localization of sensor nodes is attained with the help of two phases: ranging and position estimation phase. In first phase, each target node measures its distance from the anchor nodes using the intensity of received signal or the signal propagation time. Accurate distance measurement is not possible because of noise. In second phase, the location of the sensor nodes is estimated using the information obtained from the ranging phase. It can be done by using either geometric approach or optimization algorithm.

In the past, several interesting approaches have been used to tackle the problem of WSN node localization. In [9], a detailed survey of various localization systems for ubiquitous computing is presented. In [5], different localization techniques along with a detailed study of various measurement techniques for WSNs are described. In [10], a novel localization scheme in which all anchor nodes flood their location position to all target nodes present in the network. In this technique, every target node estimates its position by making the use of position of three or more anchor nodes. Further, in [11], an enhancement to the same technique is proposed, in which each target node makes use of the neighbors' location to improve their location accuracy. In [12], Kalman filter-based least-square estimation technique is used to address the issue of error accumulation. In [13], convex optimization-based on semi-definite programming is used to address the node localization problem in WSN. In [14] and [15], the semi-definite programming approach is extended to non-convex inequality constraints and to a gradient-search, respectively.

The optimization algorithms are really good in solving optimization problems like decision subset sum problem, feature selection, traveling salesman problem. Localization issue can be considered as an unconstrained optimization problem and can be approached with the optimization algorithms [16]. The analytical methods of optimization are not efficient in solving the localization problem because of time and complexity factors [17]. It motivated the researchers to use effective and robust nature-inspired metaheuristic algorithms [18] to solve the issue. These algorithms are inspired from the nature and help in solving various optimization problems by keeping the perfect balance among its components. The algorithms like genetic algorithm (GA), particle swarm optimization (PSO) [19], firefly algorithm (FA) [20] have been used to locate the position of the sensor nodes. There are various optimization algorithms available which can help to minimize the localization error and localize the maximum number of target nodes [21].

The primary contributions of this paper are the proposal of butterfly optimization algorithm for node localization in the WSN. A comparative analysis of the performance of BOA with FA and PSO is presented. The simulation results demonstrate that the BOA-based node localization scheme is better in terms of computing time and accuracy. The rest of this paper is organized as follows. In Sect. 2, brief review of nature-inspired metaheuristics is presented and in Sect. 3, iterative node localization is discussed. In Sect. 4, simulations and results obtained using BOA, FA and PSO are presented. Section 5 presents the concluding remarks and outlines direction for further research.

2 Nature-Inspired Metaheuristic Algorithms: A Brief Review

In various disciplines of engineering, real-world problems are formulated as optimization problems. In the past, these optimization problems are tackled by traditional methods; however, these problems require huge computational efforts, which increase with the increase in problem size. This motivated researchers to use optimization methods, which produce better results and use less computational resources [22]. Researchers have used nature-inspired metaheuristic algorithms as computationally better alternatives to traditional methods [23].

Examples of nature-inspired metaheuristic algorithms include particle swarm optimization (PSO) algorithm [24], genetic algorithm (GA) [25], butterfly optimization algorithm (BOA) [26], firefly algorithm (FA) [27,28], surrogate based optimization (SBO) [29] and many more [22,30]. Various hybrid algorithms have been developed by the researchers to improve the solution quality and convergence [31,32].

However, these optimization algorithms may not work in the best possible manner with resource-constrained computational units, like wireless sensor nodes, because of some additional computational overheads. So, BOA and the variants of PSO and FA are employed for WSNs node localization in this study. The underlying reason behind the selection of these algorithms is the ability of these algorithms to produce better results when applied to various real-world problems. Moreover, FA and PSO are easy to implement and have good convergence rate, while BOA produces superior quality of solutions and uses very less memory.

2.1 Butterfly Optimization Algorithm

Butterfly optimization algorithm (BOA) is a new nature-inspired metaheuristic algorithm developed by Arora [26]. It is based on food-foraging strategy of butterflies. Butterflies use sense receptors to locate the source of their food/nectar. These sense receptors, also called chemoreceptors, are able to sense fragrance and are scattered all over butterfly's body parts. In BOA, these butterflies are the search agents which perform optimization. In this algorithm, it is assumed that every butterfly generates fragrance having some intensity and this fragrance is propagated and sensed by other butterflies in the region. The emitted fragrance of the butterfly is correlated with the fitness of the butterfly. This means when a butterfly changes its position, its fitness/fragrance will vary accordingly [26]. When a butterfly senses greater amount of fragrance emitted by some other butterfly in the region, that particular butterfly will move toward that latter butterfly and this phase is termed as global search. In another scenario, when a butterfly is not able to sense fragrance greater than its own fragrance, it will move randomly and this phase is termed as local search phase.

The main strength of BOA lies in its mechanism to modulate fragrance in the algorithm. In order to understand the modulation, first it should be discussed that how any sense like sound, smell, heat, light is processed by a stimulus of a living organism. The whole concept of sensing and processing the modality is based on three important terms viz. sensory modality (c), stimulus intensity (I) and power exponent (a). Sensory modality is the concept related to measuring the form of energy and processing it. Stimulus intensity is the magnitude of the physical/actual stimulus. In BOA, the stimulus intensity is correlated with the fitness of the butterfly/solution. This means that when a butterfly is emitting greater amount of fragrance, the other butterflies in that surrounding can sense it and gets attracted toward it [33]. Power is the exponent to which the intensity is raised. The natural phenomenon of butterflies is based on two important issues: the variation of I and formulation of f. For simplicity, I of a butterfly is associated with encoded objective function. However, f is relative i.e., it should be sensed by other butterflies. Using these concepts, in BOA, the fragrance is formulated as a function of the physical intensity of stimulus as follows:

$$f = cI^a \tag{1}$$

where f is the perceived magnitude of the fragrance, i.e., how stronger the fragrance is perceived by other butterflies, 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.

There are two key steps in the algorithm, they are global search phase and local search phase. In global search phase, the butterfly takes a step toward the most fittest butterfly/solution g^* which can be represented as:

$$x_i{}^{t+1} = x_i{}^t + \text{Lévy}(\lambda) \times (g^* - x_i{}^t) \times f_i \tag{2}$$

where $x_i{}^t$ is the solution vector x_i for ith butterfly in iteration number t. Here, g^* represents the current best solution found among all the solutions in the current iteration. F_i represents the fragrance of ith butterfly, and λ is the step size.

Local search phase can be represented as:

$$x_i{}^{t+1} = x_i{}^t + \text{Lévy}(\lambda) \times (x_j{}^t - x_k{}^t) \times f_i \tag{3}$$

where x_j and x_k are randomly chosen butterflies from the solution space. If x_j and x_k belong to the sub-swarm and λ is the step size, then Eq. (4) becomes a stochastic equation for random walk.

$$L\acute{e}vy \sim u = t^{-\lambda}, \qquad (1 < \lambda <= 3), \tag{4}$$

The steps of butterfly essentially form a random walk process according to power-law step-length distribution with a heavy tail which has an infinite variance with an infinite mean. The use of Lèvy flights in the movement of butterflies speeds up the local search by generating new solutions around the best solutions generated so far. However, some solutions should be generated by far fields randomization and positions of those solutions should be distant from current best solution which makes sure that solutions will not be trapped in local optima.

Search for food and mating partner by butterflies can occur at both local and global scales. Considering physical proximity and various other factors like rain, wind, search for food can have a significant fraction p in overall food- or mating partner-searching activities of butterflies. So a switch probability p is used in BOA to switch between common global search to intensive local search. The above two key steps plus the switch condition can be summarized in the pseudo code shown in Algorithm 1.

Algorithm 1 Pseudo code of the Butterfly Optimization Algorithm (BOA)

```
1: Objective function $f(\mathbf{x})$, $\mathbf{x}=(x_1\ldots\ldots x_{dim})$
2: Generate population of $n$ Butterflies $\mathbf{x}_i= (i=1,2,\ldots n)$
3: Define $c$, $a$ and $p$
4: while stopping criteria not met do
5:     for each butterfly $bf$ in population do
6:         Calculate fragrance for $bf$ using Eq. (1)
7:     end for
8:     Find the best $bf$
9:     for each butterfly $bf$ in population do
10:        Generate a random number $r$ from [0, 1]
11:        if $r < p$ then
12:            Move towards best butterfly using Eq. (2)
13:        else
14:            Move randomly using Eq. (3)
15:        end if
16:    end for
17: end while
18: Output the best solution found.
```

2.2 Firefly Algorithm

Firefly algorithm (FA) is a nature-inspired metaheuristic algorithm which mimics the social behavior of fireflies found in the tropical region. Basically, fireflies produce several types of flashing patterns in order to communicate, search and find their mating partner. These flashing characteristics of fireflies were idealized by Yang [27] in order to develop a firefly inspired algorithm. In FA, three rules were idealized, which are:

(i) All the fireflies are assumed as unisexual by which any firefly can get attracted toward other firefly present in the surrounding, irrespective of their sex.
(ii) Attractiveness of each firefly is directly proportional to their brightness. It means any firefly with less brightness will move toward that butterfly which displays more brightness.
(iii) The brightness of a firefly is calculated using the objective function.

The main algorithm of FA is focused on two important issues, i.e., how the light intensity is to be varied and how the attraction is formulated. For easiness, the attractiveness of each firefly is calculated by its brightness which is further correlated with the determined objective function. In a general case of maximization problem, at a particular position the brightness I of a firefly can be assumed as $I(x) \propto f(x)$. However, the attractiveness β is relative which means it should be visualized by other fireflies in the region. Therefore, it should differ with the change in the distance between the fireflies. According to the basic laws of physics, the light intensity and henceforth attractiveness should decrease with the increase in the distance from the source, which means the modification of light intensity and attractiveness should be monotonically decreasing functions. The combined effect of both the inverse square law and absorption can be defined as:

$$I = I_0 \exp(-\gamma r^2) \tag{5}$$

where I is the light intensity, I_0 is the initial light intensity and γ is the coefficient which accounts for varying degree of light absorption factors. The attractiveness of a firefly is proportional to the light intensity visualized by other fireflies in the region, so the attractiveness β can be defined as:

$$\beta = \beta_0 \exp(-\gamma r^2) \tag{6}$$

where β_0 is the attractiveness at distance $r = 0$ and γ is light absorbtion coefficient. The distance r_{ij} between any two fireflies ith and jth located at X_i and X_j, respectively, is determined using the Euclidean norm, and movement of a less brighter firefly ith toward brighter firefly jth is determined by

$$x_i = x_i + \beta_0^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha\left(\text{rand} - \frac{1}{2}\right) \tag{7}$$

where the second term is due to the attraction and third term is a randomization with the vector of random variable. The basic steps of FA are summarized as the pseudo code shown in Algorithm 2.

Algorithm 2 Pseudo code of Firefly Algorithm (FA)

```
1: Objective Function f(X), X= ($x_1$, $x_2$,…$x_d$)
2: Generate population of n fireflies, $X_i$, $i$ = 1,2,…,n
3: Light intensity $I_i$ at $X_i$ is determined by f($X_i$)
4: Define the light absorption coefficient $\gamma$
5: while ($t$ < MaxGeneration)
6:    for $i$ = 1: $n$, all $n$ fireflies
7:       for $j$ = 1: $n$, all $n$ fireflies (inner loop)
8:          if ($I_i$ < $I_j$),
9:             Move firefly $i$ towards $j$ using Eq. (7)
10:         end if
11:         Vary attractiveness with distance $i$ via exp[- $\gamma r^2$]
12:      end for $j$
13:   end for $i$
14:   Rank the fireflies and find the global best solution
15: end while
16: Post-process the results
```

2.3 Particle Swarm Optimization

Particle swarm optimization (PSO) algorithm is a swarm intelligence-based algorithm which simulates the social behavior of bird flocks [34]. PSO is among the most powerful algorithms for optimization. The PSO algorithm employs a set of individuals which populated in the search space with random initial locations. Each individual, i, in a particle swarm is composed of three vectors, with a dimensionality equal to that of the problem space. These are the current position, previous best position and associated velocity denoted as x_i, p_i, and v_i, respectively. The position, x_i, represents a set of coordinates describing a point in solution space. In each iteration, the current position is evaluated as a problem solution and, if the particle finds better result than its previous best one, then it substitutes the one stored in p_i. The best fitness result found by each particle is stored in pbest$_i$, whereas the best result encountered by the entire population is stored in gbest$_i$ to use it for comparisons in later iterations. Let $x_i = (x_{i1}, x_{i2} \dots x_{iN})$ be the N-dimensional vector representing the position of the ith particle in the swarm, pbest $= [p_1, p_2, \dots p_N]$ be the position vector of the ith particle's personal best, gbest $= [g_1, g_2, \dots g_N]$ be the position vector of the best particle in the swarm and $v_i = [v_{i1}, v_{i2} \dots v_{iN}$ be the velocity of the ith particle. The movement of the particle is mathematically modeled as:

$$v_{id} = wv_{id} + c_1r_1(p_{id} - x_{id}) + c_2r_2(g_d - x_{id}) \tag{8}$$

$$x_{id} = x_{id} + v_{id} \tag{9}$$

where $i = 1, 2, \dots K; d = 1, 2, \dots N$; where K represents the population size of the swarm. w is the inertial weight, and c_1 and c_2 are the cognitive and social scaling parameters, respectively. r_1 and r_2 represent random numbers drawn from a uniform range of [0, 1]. The parameters c_1 and c_2 play vital role in the convergence characteristics of PSO as c_1 determines how much a particle is influenced by the memory of its best solution, whereas c_2 determines the impact of swarm on the particle. Another important parameter is w as its small value will result in premature convergence and larger value will result in slow convergence. The basic steps of the PSO algorithm can be summarized as the pseudo code shown in Algorithm 3.

Algorithm 3 Pseudo code of Particle Swarm Optimization (PSO) algorithm

```
1: Objective Function f (X), X= ($x_1$, $x_2$,…$x_d$)
2: Generate initial population of n particles, $X_i$,$i$ =1,2,.,n
3: while ($t$< MaxGeneration)
4: for each particle
5:       Calculate fitness value
6:        Update best fitness value (pBest) in history
7:         Set current value as the new pBest
8:    end for
9:   Choose the particle with the best fitness value as gBest
10:   for each particle
11:        Calculate particle velocity according Eq. (8)
12:        Update particle position according Eq. (9)
13:   end for
14: end while
15: Post-process the results
```

3 Iterative Node Localization

The distributed range-based localization technique is used to estimate the coordinates of sensor nodes. The main objective of node localization in WSN is to determine the coordinates of the most of the target nodes by minimizing the objective function. The localization issue of WSN is considered as an optimization problem which is approached by various metaheuristic algorithms. The following process is used to localize the sensor nodes in WSN:

1. Initialize M target nodes and N anchor nodes randomly in the sensor field. Each anchor node has location awareness to find its location. Each anchor node and target node has transmission range R.
2. The distance between each target node and anchor nodes is evaluated which is altered by the additive Gaussian noise. Each target node calculates the distance by using equation $\hat{d_i} = d_i + n_i$ where d_i is the real distance which is calculated between the position of the target node (x, y) and the position of the beacon (x_i, y_i) by using the following equation :

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{10}$$

   The variable n_i is the noise affecting the measured distance distributed in the range $d_i \pm d_i\left(\frac{P_n}{100}\right)$ where P_n is the percentage of noise in measured distance.
3. The target node is known as localizable node if there are at least three anchor nodes within the transmission range of the target node. The underlying reason behind this requirement is that according to the trilateral positioning method, the coordinates of the three anchor nodes A (x_1, y_1), B (x_2, y_2), and C (x_3, y_3), and the distance between the target node d_i and three anchor nodes are known. Then, by using the trigonometric laws of sines or cosines, the coordinates of the target node are calculated. Similarly, in multilateration target node estimation method, distance measurements of three or more anchor nodes are used to minimizing the error between actual distance and estimated distance. The method of calculation can be seen from Fig. 1.
4. For each localizable node, metaheuristic algorithm is run independently to find the position of the target node. The butterflies or agents are initialized with the centroid of the anchor nodes that are within transmission range by:

$$(x_c, y_c) = \left(\frac{1}{N}\sum_{i=1}^{N} x_i, \frac{1}{N}\sum_{i=1}^{N} y_i\right) \tag{11}$$

   where N is the total number of anchor nodes within transmission radius of the localizable target node.

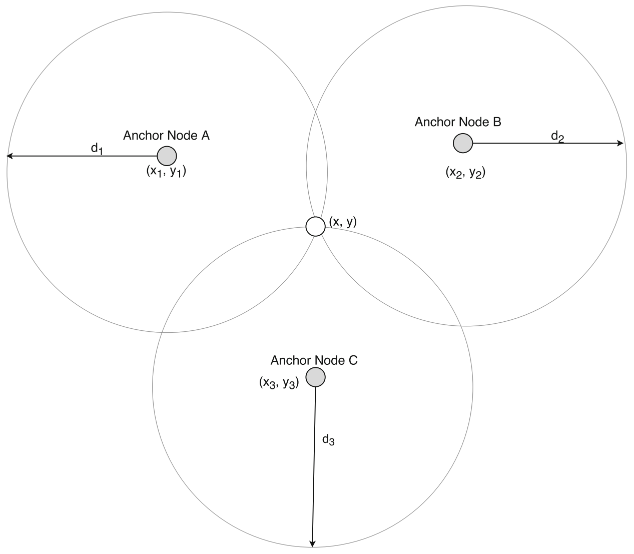

Fig. 1 Principle of trilateral positioning method

5. The metaheuristic algorithm helps to find the coordinates (x, y) of the target node that minimizes the localization error. The objective function of localization problem is the mean square distance between target node and anchor node which is minimized using an algorithm and it is described mathematically as:

$$f(x, y) = \frac{1}{N}\left(\sum_{i=1}^{N} \sqrt{(x - x_i)^2 + (y - y_i)^2} - \hat{d_i}\right)^2 \tag{12}$$

   where $N \geq 3$ is the number of anchor nodes within transmission range of the target node.
6. The optimal value (x, y) is estimated by metaheuristic algorithm after number of generations by minimizing the objective function.
7. The total localization error is computed after the position of all localizable target nodes N_{L} is estimated. It is calculated as the mean of square of the distance between the estimated node coordinates (X_i, Y_i) and the actual node coordinates (x_i, y_i) which is given as:

$$E_1 = \frac{1}{N_1}\sum_{i=1}^{N} \sqrt{(x_i - X_i)^2 + (y_i - Y_i)^2} \tag{13}$$

8. The steps from 2 to 6 are iterated until all the target nodes get localized or no more nodes can be localized. The localization algorithm's performance depends on the average localization error E_1 and the number of unlocalized nodes $N_{N_{\mathrm{L}}}$ which can be evaluated by using the equation $N_{N_{\mathrm{L}}} = M - N_{\mathrm{L}}$. The smaller values of E_1 and $N_{N_{\mathrm{L}}}$ make the localization more efficient.

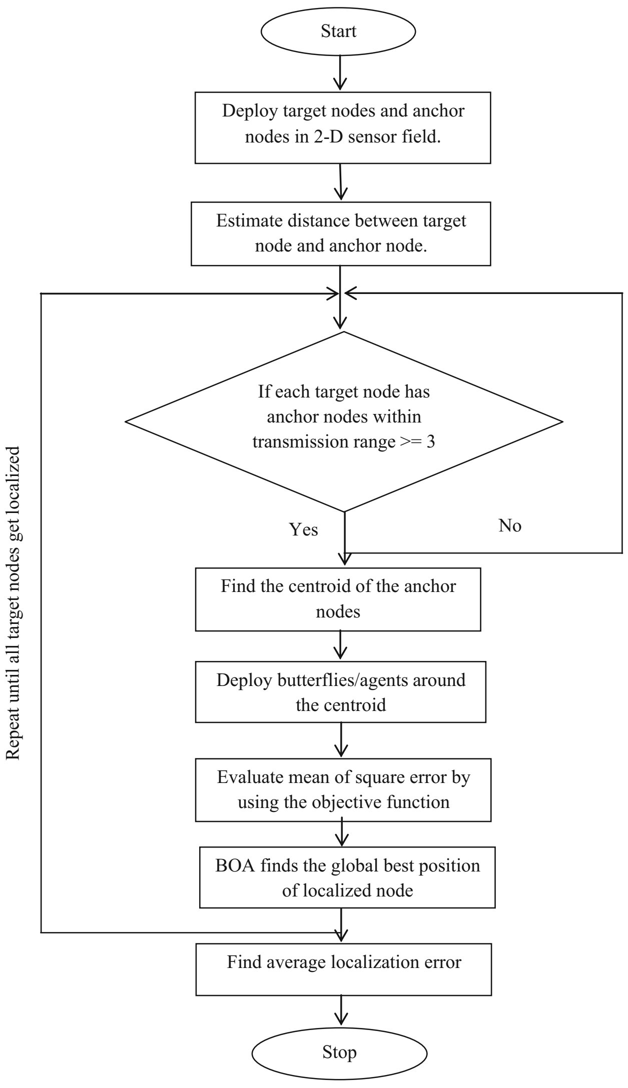
Start
Deploy target nodes and anchor nodes in 2-D sensor field.
Estimate distance between target node and anchor node.
If each target node has anchor nodes within transmission range >= 3
Yes
No
Find the centroid of the anchor nodes
Deploy butterflies/agents around the centroid
Evaluate mean of square error by using the objective function
BOA finds the global best position of localized node
Find average localization error
Stop
Repeat until all target nodes get localized

Fig. 2 Flowchart for BOA-based node localization in WSN

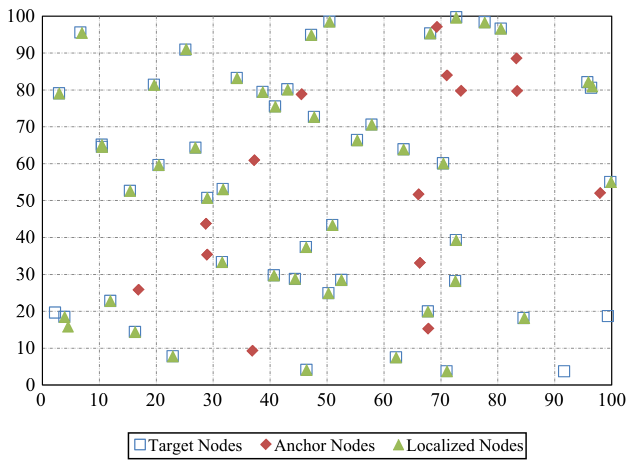

Fig. 3 Node localization using BOA

The number of localized nodes increases as the iteration progresses. It also increases the number of anchor nodes within the transmission range of the localizable target node as the estimated position of target node behaves as a anchor node in the next iteration. It helps to reduce the problem of flip ambiguity which produces large localization error. However, computation time to obtain localization information of the target node increases as the iteration increments. The overall flowchart is shown in Fig. 2.

4 Numerical Simulation and Results

The simulations and performance analysis of the proposed node localization scheme are conducted in QT Creator 2.4.0. For simulations, a sensor network with static target and anchor nodes is deployed in a 100 m × 100 m area. As shown in Fig. 3, the positions of the sensor nodes are represented by random generated coordinates (x, y) within the boundary. The total number of target nodes in the given area is fixed to 50 for the simulations. Any particular ranging technique is not considered in this study. In these simulations, it is assumed that the range measurement is blurred with additive white gaussian noise only, i.e., $\hat{d_i} = d_i + n_i$ as mentioned in Eq. (10) where n_i is zero mean gaussian variable with variance $\sigma^2{}_d$. The standard deviation of the measured distance σ_d is the parameter affecting the performance of the localization. The density of anchor nodes (per m^2) and the transmission range of sensor nodes are the important parameters influencing the localization error. Each point in the simulation result is the average of 30 replication and is plotted with a confidence interval of 95%. The transmission range of anchor nodes is set as 30 U. The size of population n and number of generations are fixed to 30 and 200, respectively. Similar techniques have been used by various researchers in the past [7,18,35]. The strategic settings and parameter values of BOA, PSO and FA are discussed below:

4.1 Case Study 1: Node Localization Using BOA

In this case study, each localized node runs BOA to estimate the location. In BOA, the initial value of modular modality c is taken as 0.01, whereas the initial value of power exponent a is set to 0.1. In this study, pseudorandom numbers are used instead of lévy flights. The reason behind choosing the pseudorandom numbers over lévy flights is that it increases the convergence rate as well the chances to gain the global optimality. The localization of 50 target nodes using BOA is depicted in Fig. 3.

4.2 Case Study 2: Node Localization Using FA

In this case study, each localized node runs FA to estimate the location. In FA, the value of randomization parameter α is taken as 0.25, whereas the absorption coefficient γ is set

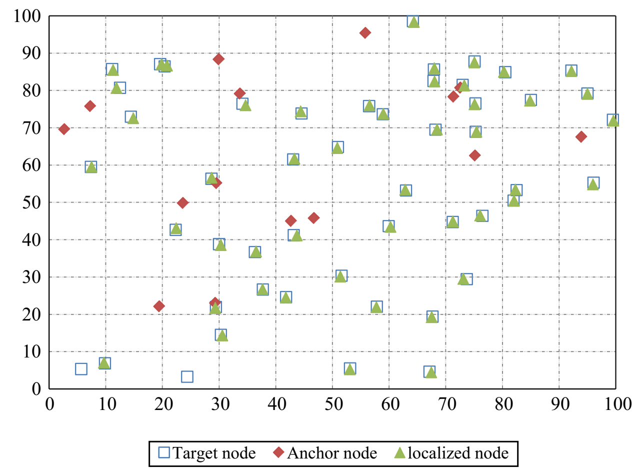

Fig. 4 Node localization using FA

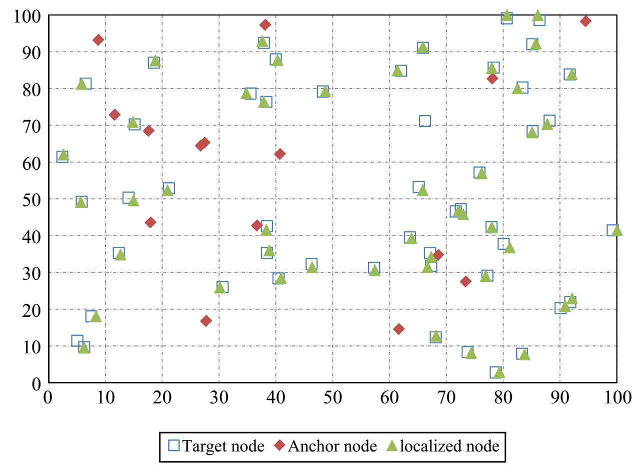

Fig. 5 Node localization using PSO

to 1.0. The initial attractiveness parameter β is set to 1 [36]. FA is run for each target node till the maximum number of generations to localize target nodes. FA-based localization for 50 target nodes is represented in Fig. 4.

4.3 Case Study 3: Node Localization Using PSO

For PSO, initial values of $w = 0.7$ and $c_1 = c_2 = 1.494$ were recommended for faster convergence after experimental tests [37,38]. In this study, gbest PSO algorithm with adaptable weight methods is used [39]. The PSO algorithm is executed using above parameters for specified number of iterations to find the optimal value. The localization of 50 target nodes using PSO is depicted in Fig. 5.

Results of BOA-, FA- and PSO-based localization summarized in Table 1 demonstrate that all the algorithms used here have performed fairly well in WSN localization. The effect of P_n, percentage noise in distance measurement, on localization accuracy is evident. Average localization error in all the algorithms is reduced for $P_n = 2$. The performance metric doublet (N_{NL}, E_1) for BOA is less than that for FA and PSO, thus indicting superior performance of BOA. Moreover, the computing time required for BOA is also significantly less than that for FA and PSO.

The detailed observations made in the five trial runs out of the 30 experiments are summarized in Table 2. As shown in Table 2, the number of localized nodes is represented by N_{L}, in each iteration. The proposed localization scheme is stochastic, so same results are not produced in all runs or experiments. Due to this, the results of various trial experiments are averaged and are summarized in Table 1. According to the simulation results demonstrated in Tables 1 and 2, the proposed localization scheme using BOA performs better in terms of localization error and un-localized nodes. PSO performs worst, but its computation time is less than all other algorithms used in this study. The initial deployment of sensor nodes is random due to which the localization accuracy and the total computing time may vary in different trials. The anchor nodes, target nodes and the position estimated by the algorithms like BOA, FA and PSO are shown in Figs. 3, 4 and 5. The critical parameters which effect the localization error of nodes are number of anchor nodes, transmission range and number of iterations of optimization algorithms.

4.4 Effect of Anchor Node Density

Location estimation accuracy and the number of localized nodes increase with the increase in anchor node density. It is difficult to locate position of nodes if sufficient number of anchor nodes ($N \geq 3$) are not available. The performance of the localization algorithm depends on the density of anchor nodes. A less number of anchor nodes localize very few number of target nodes. The percentage of the localized nodes depends on the number of anchor nodes for BOA, PSO and FA as shown in Fig. 6. The percentage of localized node increases with an increase in number of anchor nodes.

4.5 Effect of Transmission Range

The increase in transmission range of anchor nodes helps in improving the performance as the number of anchor nodes within range will be more. This will also increase the number of localized nodes. The percentage of localized nodes relies on the transmission range for BOA, PSO and FA as shown in Fig. 7. According to the simulation results, it can be analyzed that using smaller transmission range will result in localization of very less number of sensor/target nodes. Gaussian additive noise is also an important parameter that really affects the localization accuracy. As noise in distance measurement increases, E_1 increases which leads to decrease in localization accuracy. Due to this, all the experiments are conducted by considering Gaussian noise $P_n = 2$. The local-

Table 1 Summary of results of 30 trial runs of BOA-, FA- and PSO-based node localization

Algorithms	$P_n = 5$			$P_n = 2$		
	Mean N_{N_L}	Mean E_L	Computing time (in secs)	Mean N_{N_L}	Mean E_L	Computing time (in secs)
BOA	4.7	0.28	0.65	4.5	0.21	0.53
FA	6.6	0.72	2.15	6.2	0.69	1.94
PSO	5.9	0.81	0.54	5.6	0.78	0.49

$M = 40$, $N = 8$, and $r = 30$ U and the sensor field size = 100 × 100 square units
* All the experiments are conducted on the same computer

Table 2 Summary of results of BOA-, FA- and PSO-based node localization

Target node	Anchor node	Trial	BOA			FA			PSO		
			N_L	E_l	T_l	N_L	E_l	T_l	N_L	E_l	T_l
25	10	1	23	0.207908	0.40	19	0.335551	1.44	22	0.807158	0.36
		2	24	0.188224	0.33	20	0.246423	1.44	17	0.728214	0.39
		3	25	0.224510	0.38	21	0.296398	1.70	18	0.797650	0.40
		4	25	0.19963	0.38	20	0.256168	1.65	17	0.739102	0.39
		5	24	0.212247	0.31	19	0.278459	1.57	19	0.799164	0.36
50	15	1	46	0.235326	0.77	50	0.505511	2.50	48	0.578797	0.74
		2	49	0.260490	0.81	49	0.326980	4.42	50	0.753254	0.85
		3	48	0.361080	0.92	49	0.254824	1.63	47	0.587004	0.75
		4	50	0.323910	0.86	48	0.227842	3.90	46	0.438748	0.76
		5	49	0.351415	0.91	49	0.2476413	4.19	47	0.486784	0.85
75	20	1	75	0.328310	1.68	74	0.703964	2.97	75	0.67414	1.31
		2	75	0.219680	1.52	75	0.291862	2.73	75	0.720123	1.35
		3	68	0.178960	1.52	72	0.279126	5.84	73	0.771325	1.30
		4	75	0.183942	1.43	71	0.284865	4.70	72	0.798457	1.32
		5	73	0.196781	1.69	73	0.2907846	3.97	73	0.697814	1.31
100	25	1	100	0.218838	2.29	100	0.779716	5.66	100	0.668227	2.49
		2	100	0.295008	2.27	100	0.299194	6.33	100	0.614843	2.10
		3	100	0.216414	2.32	100	0.385758	3.55	100	0.608155	2.20
		4	100	0.235804	2.37	100	0.589494	4.56	100	0.627197	2.35
		5	100	0.259312	2.45	100	0.513591	4.93	100	0.653258	2.16
125	30	1	124	0.615712	3.12	122	0.938894	2.707	119	0.600957	3.90
		2	123	0.437651	3.65	123	0.651459	5.995	123	0.662322	3.87
		3	125	0.568754	4.26	123	0.831683	2.709	125	0.593421	4.90
		4	124	0.657499	3.76	125	0.950842	3.11	125	0.608412	3.98
		5	125	0.545789	3.87	125	0.912666	5.894	125	0.744193	4.90
150	35	1	150	0.743780	5.67	149	0.957818	3.386	149	0.657679	5.03
		2	150	0.887561	4.87	150	0.973891	3.459	149	0.773764	5.16
		3	150	0.765347	5.65	150	0.854096	5.894	150	0.620403	4.22
		4	149	0.665348	4.12	150	0.672451	4.87	150	0.766621	5.21
		5	150	0.787689	4.76	150	0.632727	3.356	150	0.625278	4.43

N_L = number of localized nodes E_l = localization error T_l = computing time (in seconds)

ization accuracy also improves with the increase in number of iterations as shown in Fig. 8. As the number of iteration progresses, the localization error declines. BOA shows more decline in the error as compared to other two algorithms.

4.6 Effect of Number of Iterations

The increase in number of iterations helps in localizing more number of nodes. This increases the number of references

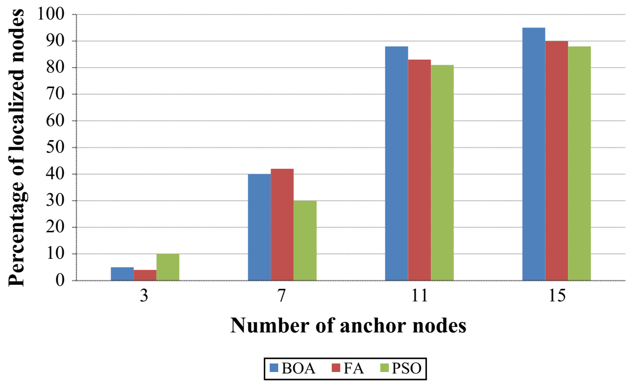

Fig. 6 Percentage of localized nodes versus number of anchor nodes

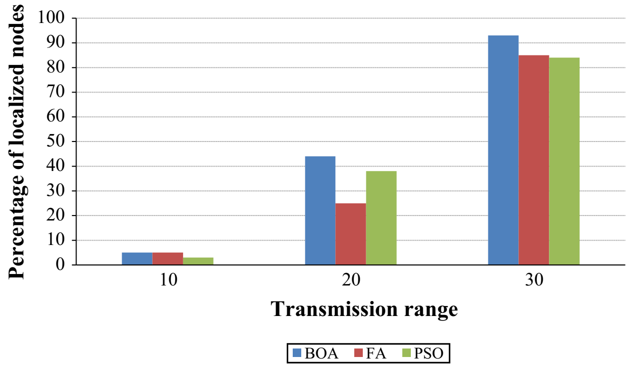

Fig. 7 Percentage of localized nodes versus transmission range

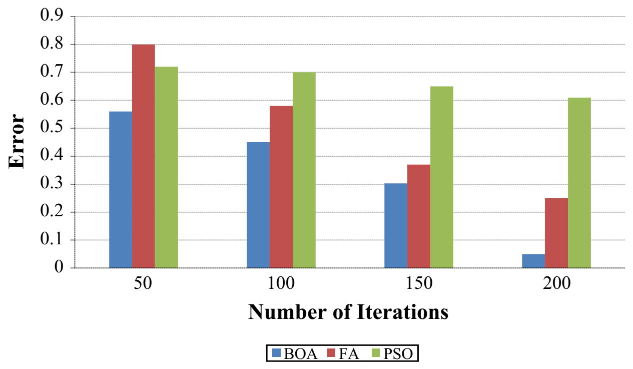

Fig. 8 Error versus number of iterations

available for already localized nodes, which further decreases the probability of the flip ambiguity. On the other hand, if a node has more references in iteration $k + 1$ than in iteration k, the time required for localization increases. This claim is supported by our simulation results shown in Fig. 8. It can be seen from Fig. 8 that the localization accuracy improves with the increase in number of iterations. Particularly, in the case of BOA, there is significant decline in the error as compared to other two algorithms.

BOA-based node localization estimated the location of nodes with minimum localization error. Node localization results based on BOA, PSO and FA algorithms by varying number of anchor and target nodes are summarized in Table 2. In order to better investigate the performance of proposed algorithm, all the experiments are conducted with different configurations. The BOA-based localization algorithms provide less localization error in estimating the target nodes, whereas PSO estimates the position in less computing time, but it has high localization error. All the optimization algorithms performed well to determine the location of nodes in WSN. BOA has better localization accuracy to estimate the position than FA and PSO in terms of mean square error.

5 Conclusion and Future Work

Localization of sensor nodes is really important for the performance of WSN as many applications of WSN require localization information. The main objective of this optimization problem is to minimize the localization error with the help of nature-inspired optimization algorithms. In this paper, BOA-based node localization algorithm is proposed to estimate the position of the sensor nodes in WSN. This paper has described the BOA-based localization technique and provides the summary of results by comparing the algorithm with the others like PSO algorithm and FA in terms of localization error, localized nodes and computing time. The simulation results show that the proposed technique is an effective refinement technique in nodes localization. BOA clearly outperforms other algorithms used in this study in terms of accuracy and computing time.

Future work will investigate the performance of the proposed method for centralized method and distributed method to solve the energy issues in WSN. Further, BOA can be hybridize with other optimization algorithm to further minimize the location estimation error.

References

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E.: A survey on sensor networks. IEEE Commun. Mag. **40**(8), 102–114 (2002)
2. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E.: Wireless sensor networks: a survey. Comput. Netw. **38**(4), 393–422 (2002)
3. Kulkarni, R.V.; Förster, A.; Venayagamoorthy, G.K.: Computational intelligence in wireless sensor networks: a survey. IEEE Commun. Surv. Tutor. **13**(1), 68–96 (2011)
4. Yick, J.; Mukherjee, B.; Ghosal, D.: Wireless sensor network survey. Comput. Netw. **52**(12), 2292–2330 (2008)
5. Mao, G.; Fidan, B.; Anderson, B.D.: Wireless sensor network localization techniques. Comput. Netw. **51**(10), 2529–2553 (2007)
6. Wang, J.; Ghosh, R.K.; Das, S.K.: A survey on sensor localization. J. Control Theory Appl. **8**(1), 2–11 (2010)
7. Aspnes, J.; Eren, T.; Goldenberg, D.K.; Morse, A.S.; Whiteley, W.; Yang, Y.R.; Anderson, B.; Belhumeur, P.N.: A theory of network localization. IEEE Trans. Mob. Comput. **5**(12), 1663–1678 (2006)
8. Patwari, N.; Ash, J.N.; Kyperountas, S.; Hero III, A.O.; Moses, R.L.; Correal, N.S.: Locating the nodes: cooperative localization

in wireless sensor networks. IEEE Signal Process. Mag. **22**(4), 54–69 (2005)
9. Hightower, J.; Borriello, G.: Location systems for ubiquitous computing. Computer **8**, 57–66 (2001)
10. Niculescu, D.; Nath, B.: Ad hoc positioning system (APS). In: Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE, vol. 5, pp. 2926–2931 (2001)
11. Rabaey, C.S.J.; Langendoen, K.: Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In: USENIX Technical Annual Conference, pp. 317–327 (2002)
12. Savvides, A.; Park, H.; Srivastava, M.B.: The n-hop multilateration primitive for node localization problems. Mob. Netw. Appl. **8**(4), 443–451 (2003)
13. Doherty, L.; Pister, K.S.; El Ghaoui, L.: Convex position estimation in wireless sensor networks. In: INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings, vol. 3, pp. 1655–1663 (2001)
14. Biswas, P.; Lian, T.-C.; Wang, T.-C.; Ye, Y.: Semidefinite programming based algorithms for sensor network localization. ACM Trans. Sens. Netw. (TOSN) **2**(2), 188–220 (2006)
15. Liang, T.-C.; Wang, T.-C.; Ye, Y.: A gradient search method to round the semidefinite programming relaxation solution for ad hoc wireless sensor network localization. Sanford University, formal report 5 (2004)
16. Yun, S.; Lee, J.; Chung, W.; Kim, E.; Kim, S.: A soft computing approach to localization in wireless sensor networks. Expert Syst. Appl. **36**(4), 7552–7561 (2009)
17. Harikrishnan, R.; Kumar, V.J.S.; Ponmalar, P.S.: A comparative analysis of intelligent algorithms for localization in wireless sensor networks. Wirel. Pers. Commun. **87**(3), 1057–1069 (2016)
18. Kulkarni, R.V.; Venayagamoorthy, G.K.; Cheng, M.X.: Bio-inspired node localization in wireless sensor networks. In: IEEE International Conference on Systems, Man and Cybernetics, 2009. SMC 2009, pp. 205–210 (2009)
19. Gopakumar, A.; Jacob, L.: Localization in wireless sensor networks using particle swarm optimization. In: IET International Conference on Wireless, Mobile and Multimedia Networks, 2008, pp. 227–230 (2008)
20. Harikrishnan, R.; Kumar, V. J. S. and Ponmalar, P. S.: "Firefly algorithm approach for localization in wireless sensor networks," in *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*, pp. 209–214, Springer, Berlin (2016)
21. Boukerche, A.; Oliveira, H.A.; Nakamura, E.F.; Loureiro, A.A.: Localization systems for wireless sensor networks. IEEE Wirel. Commun. **14**(6), 6–12 (2007)
22. Vasant, P.: Handbook of research on artificial intelligence techniques and algorithms, vol. 2. Information Science Reference-Imprint of IGI Publishing (2015)
23. Del Valle, Y.; Venayagamoorthy, G.K.; Mohagheghi, S.; Hernandez, J.-C.; Harley, R.G.: Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Trans. Evolut. Comput. **12**(2), 171–195 (2008)
24. Kulkarni, R.V.; Venayagamoorthy, G.K.: Particle swarm optimization in wireless-sensor networks: a brief survey. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **41**(2), 262–267 (2011)
25. Schaefer, R.: Foundations of Global Genetic Optimization. Springer, Berlin (2007)
26. Arora, S.; Singh, S.: Butterfly algorithm with l'evy flights for global optimization. In: 2015 International Conference on Signal Processing, Computing and Control (2015 ISPCC) (2015)
27. Yang, X.-S.: Firefly algorithm, stochastic test functions and design optimisation. Int. J. Bio-Inspir. Comput. **2**(2), 78–84 (2010)
28. Cao, S.; Wang, J.; Gu, X.: A wireless sensor network location algorithm based on firefly algorithm. In: AsiaSim 2012, pp. 18–26. Springer, Berlin (2012)
29. Al-Adwani, S.; Elkamel, A.; Duever, T.A.; Yetilmezsoy, K.; Abdul-Wahab, S.A.: A surrogate-based optimization methodology for the optimal design of an air quality monitoring network. Can. J. Chem. Eng. **93**(7), 1176–1187 (2015)
30. Yang, X.-S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press, Beckington (2010)
31. Gupta, S.; Arora, S.: A hybrid firefly algorithm and social spider algorithm for multimodal function. In: Berretti S., Thampi S., Srivastava P. (eds.) Intelligent Systems Technologies and Applications, vol 384. Springer, Cham (2016). doi:10.1007/978-3-319-23036-8_2
32. Arora, S.; Singh, S.; Singh, S.; Sharma, B.: Mutated firefly algorithm. In: 2014 International Conference on Parallel, Distributed and Grid Computing (PDGC), IEEE, pp. 33–38 (2014)
33. Arora, S.; Singh, S.: An improved butterfly optimization algorithm with chaos. J. Intell. Fuzzy Syst. **32**(1), 1079–1088 (2017)
34. Kennedy, J.: Particle swarm optimization. In: Gass, S.I., Fu, M.C. (eds.) Encyclopedia of Machine Learning, pp. 760–766. Springer, New York (2010)
35. Kulkarni, R.V.; Venayagamoorthy, G.K.: Bio-inspired algorithms for autonomous deployment and localization of sensor nodes. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **40**(6), 663–675 (2010)
36. Arora, S.; Singh, S.: A conceptual comparison of firefly algorithm, bat algorithm and cuckoo search. In: 2013 International Conference on Control Computing Communication and Materials (ICCCCM), pp. 1–4, IEEE (2013)
37. Eberhart, R.C.; Shi, Y.: Tracking and optimizing dynamic systems with particle swarms. In: Proceedings of the 2001 Congress on Evolutionary Computation, 2001, IEEE , vol. 1, pp. 94–100 (2001)
38. Shi, Y. et al.: Particle swarm optimization: developments, applications and resources. In *Proceedings of the 2001 Congress on Evolutionary Computation, 2001.*, vol. 1, pp. 81–86, IEEE (2001)
39. Parsopoulos, K.E.; Vrahatis, M.N.: Recent approaches to global optimization problems through particle swarm optimization. Nat. Comput. **1**(2–3), 235–306 (2002)