Wireless Networks (2021) 27:1999–2014
https://doi.org/10.1007/s11276-021-02557-7

An efficient localization approach to locate sensor nodes in 3D wireless sensor networks using adaptive flower pollination algorithm

Prabhjot Singh[1] · Nitin Mittal[1]

Accepted: 19 January 2021 / Published online: 8 February 2021

Abstract
Localization in wireless sensor networks (WSNs) is required to examine the coordinates of the sensor nodes deployed in the sensing field. It is the process that determines the location of the target nodes relative to the location of deployed anchor nodes. The anchor nodes positions are known as the nodes that have GPS unit incorporated with them. All sensor nodes are generally not configured with GPS as it is not suitable for indoor environments and/or underwater areas. A network becomes more expensive and utilizes more energy if all nodes are equipped with GPS that is a major drawback of WSNs. Various localization schemes have been proposed in literature, while most research proposals deal with the study of 2D applications. However, in the 3D applications, the area under observation may have a complexity in the sensing environment. An optimized algorithm is required for the determination of node location in 3D environment. In this paper, we propose an adaptive flower pollination algorithm (AFPA) with enhanced exploration and exploitation capabilities of conventional FPA for the localization of sensor nodes in WSN. To test the performance of AFPA, benchmark functions (CEC 2019) are used to compare it with other meta-heuristics. The results show that proposed AFPA outperforms in terms of convergence speed and provides better results for most of the benchmark functions. Also, the proposed AFPA is tested on WSN Localization problem, it provides least localization error in comparison to other techniques applied in 3D WSN environments.

Keywords WSN · Localization · GWO · FPA · Adaptive FPA

1 Introduction

WSN comprises of a network of sensors that are deployed either in static or mobile conditions randomly in a given geographical area to gather, process, and communicate data collected from the environment to a special node considered as the common point of contact referred as a sink [1, 2]. In WSN, sensor nodes re-locate their positions to collect the data about the sudden change occurring in the field. The information gathered by the sensors is communicated periodically or after a sudden change, depending on the nature of particular application. WSN consists of many applications like environmental monitoring, healthcare, medical research, defense and military affairs etc. Sensor nodes are unreliable and consumes more energy [3] but are cheaper and deployed everywhere in the coming future. To meet up with these requirements, the exact locations of unknown target nodes (TNs) must be known. The determination of their exact location in WSN is commonly referred to as the localization problem. There are numerous solutions to assign location to the nodes, it can be done either manually or using GPS. But assigning it manually is a complex and challenging task in WSN. It is also not feasible to use GPS feature in every node because of cost constraints in deploying sensor nodes with this feature [4]. There are various localization techniques available in the literature to detect the exact location of localized nodes. By using anchor nodes (ANs), it is possible to estimate the exact location of the TNs but AN location must be known prior and only few nodes with this feature are deployed in field due to certain cost constraints [5].

✉ Nitin Mittal
mittal.nitin84@gmail.com

Prabhjot Singh
psprabh882@gmail.com

[1] Department of Electronics and Communication Engineering, Chandigarh University, Mohali, Punjab 140413, India

The range-based algorithms determine the distance among the sensors based on received signal strength indicator (RSSI), time of arrival (ToA) and angle of arrival (AoA) [6, 7]. Similarly, the algorithms that calculates the hop counts between nodes are classified as Range free techniques. Multidimensional signalling, Distance vector hop, and adhoc positioning system provides the position of various targeted nodes with lesser infrastructure requirements.

In WSN, accurate location estimation is one of the biggest challenges. In static nodes, localization can be achieved precisely but in case of moving nodes this task is quite challenging. In this paper, we deployed a single anchor to locate all unknown TNs and by using the projection of the ANs virtually in six directions using hexagonal projection for targeting the unknown nodes using Adaptive Flower Pollination Algorithm (AFPA). As soon as the TNs drop under the range of AN; the AN itself, and three virtual nodes are selected (as minimum four ANs are required to determine 3D positions). The problem of Line of Sight (LoS) is minimised by deploying AN virtually in other six directions.

The following benefits were offered for localization using the proposed method in this paper:

- A new technique of projecting VAs employing umbrella projection in the field for determining the exact locations of deployed sensors in 3D environment is proposed using Adaptive FPA Algorithm.
- By using the concept of VA, the line of sight (LoS) issues is minimized to a greater extent.
- Flip ambiguity issues in range-based methods are also reduced.

The rest of the paper is organized as follows: Sect. 2 describes the literature review on 3D localization. Section 3, AFPA is presented in detail. In Sect. 4 statistical testing of AFPA with other meta-heuristics is presented. Section 5 deals with concept of deploying a single AN and TNs to locate unknown nodes in the deployed region. Section 6 deals with results and discussions on proposed algorithm. In Sect. 7, conclusion, limitations, and future scope are presented.

2 Literature review

WSN comprises of multiple homogeneous and/or heterogeneous sensor nodes. Each node collects the data, measures it, transmits it to the corresponding node, and forward it to the sink node. By knowing the exact location of the TNs, the data obtained and transmitted is useful and meaningful. The manual incorporation of the location information in every node is very tedious and even impossible in many applications. Generally, localization is applied in two dimensional (2D) planes. But for the localization of sensor nodes in applications like sea, forests, hilly areas and in free space, 3D algorithms are required. Although, determining 3D scenario is more complex than 2D but the algorithms used for 2D needs to be modified and applied for 3D localization with some exceptions.

Bulusu et al. [8] discussed unconstrained open-air environment using different localization algorithms for low-cost devices that does not contains GPS. They proposed an RF-based localization method in that the receiver is located using Centroid method. Graefenstein et al. [9] suggested technique that is energy efficient in which RSSI method is employed to calculate the approximate separation among the target and reference nodes deployed in the field using trilateration approach. Sumathi and Srinivasan [10] have suggested a method that requires a single AN for localizing the unknown nodes using RSSI method. The least square approach for locating the fixed TNs is used in this algorithm. Guo et al. [11] developed a perpendicular intersection (PI) mobile based approach that does not specifically map RSS distances. The measurement of the node position is carried out using the geometric PI relation. Ultra-Wide Band, and ToA technique to compute node location in 3D environment is proposed by Shi et al. [12]. By using this technique, the AN and TN separation is calculated more efficiently. Distance Vector-Hop based approach for locating sensor nodes is introduced by Wang et al. [13]. The main reason behind the failure of this algorithm is complexity and its increased cost. Xu et al. [14] introduced another enhanced 3D localization technique which adopted DV-Distance with quasi-newton optimization method for optimizing the results. By considering localization accuracy and coverage, authors further tested the effectiveness of the proposed algorithm. Li et al. [15] suggested the 3D WSN localization method based on irregular RSSI model in order to calculate the relation among DOIs and the variability in signal transmission range. Ahmad et al. [16] suggested a parametric loop-division algorithm for 3D localization, in which the deployed nodes are located in an area surrounded by a group of ANs. In this method network shrink toward the centre accurately and provides accurate localization results.

In order to get the best approximate locations of TNs various meta-heuristic optimization algorithms are used for 2D and 3D scenarios. A computationally efficient swarm intelligence approach for locating static nodes is suggested by Gopakumar and Jacob [17]. Easier implementation and low memory requirements are the key benefits of this approach. Chuang and Wu [18] use RSS ranging technique for locating sensor nodes efficiently using PSO based approach. This scheme has a higher success rate in terms

of localization. Kumar et al. [19] uses H-best PSO and BBO algorithm to predict the optimum position of the randomly deployed TNs. HPSO provides mature and fast convergence, and higher accuracy is achieved using BBO.

Suyatha and Siddappa [20] implemented a hybrid localization optimization technique using differential evolution (DE) and Dynamic Weight PSO to achieve better localization accuracy. Lower localization error, better localization accuracy and improved stability results are achieved using the proposed algorithm. Arora and Singh [21] suggested a BOA optimization algorithm to optimize the location of unknown sensor nodes. The performance of this proposed algorithm, in 2D scenarios is compared and contrasted with PSO and FA. The authors concluded that this approach outperforms in terms of convergence time and location accuracy as compared to other meta heuristics.

Due to the higher accuracy, the range-based methods are commonly used but flip ambiguity is the major drawback of range-based methods. In order to address this important challenge, several researchers proposed different strategies [22–25]. Various computational intelligence techniques were introduced [26–33] for determining the location of the moving TNs using a single anchor in WSNs. Virtual ANs were deployed at the corners of the sensing field, and this algorithm is divided into two stages. Calculations of distance were performed based on RSSI during the first stage. In the later stage, virtual ANs were assumed to locate unknown TNs with the help of anchor and virtual anchors (VAs). Centroid calculations are obtained in these stages along with optimization algorithms namely PSO, HPSO, BBO, FA, and their results show better convergence time. In this work, authors proposed the concept of single anchor and VAs. Singh and Mittal proposed a hybrid DA-FA [34] approach to locate TNs in 2D environment using single anchor, and six VAs; and their results show less localization error and faster convergence as compared with PSO, HPSO, BBO, FA. Singh and Mittal [35] proposed various NMRA variants and are applied it to 3D localization problem and their results show faster convergence and least localization error in comparison to state-of-the-art algorithms.

3 The proposed adaptive FPA algorithm

Flower Pollination Algorithm (FPA) is inspired from the natural process of pollination occurring in flowering plants [36]. In FPA, each flower stands for a feasible solution and the objective function value is regarded as the fitness value. To mimic the pollination process, two different pollination phases: Global and local pollination phase are used for each flower to mimic the pollination process with a switch probability p_s. The specific process is described as follows.

Global pollination phase: For each individual, a random number *rand* is generated. If $rand < p_s$, then global pollination should be carried out. In the global pollination process, each flower updates its position according to the following equation [34]:

$$X_i^{t+1} = X_i^t + \gamma L(\lambda)(X_{best}^t - X_i^t) \tag{1}$$

where X_i^t and X_i^{t+1} are the old and new positions of ith flower, respectively, X_{best}^t is the best flower at current iteration t, which has the best fitness value in the whole population, and γ is the scaling factor that controls the step size of global pollination. Parameter L, the Lévy flight, is used as the strength of pollination in the basic FPA, and the step size (λ) obeys the Lévy distribution:

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, (s \gg s_0 > 0) \tag{2}$$

where $\Gamma(\lambda)$ is the gamma function.

Local pollination phase: If $rand > p_s$, the local pollination process should be carried out. Flower X_i^t obtains its new position X_i^{t+1} according to the difference between its old position and the position of two neighboring flowers X_p^t and X_q^t. This process is considered as local search, and the updating equation [34] is defined as

$$X_i^{t+1} = X_i^t + r(X_p^t - X_q^t) \tag{3}$$

where r is drawn from a uniform distribution [0, 1], and it is considered as a local random walk.

After pollination is completed, the new individuals update their positions by comparing fitness values. If the fitness of X_i^{t+1} is better than that of X_i^t, the new position of i^{th} flower will be replaced by X_i^{t+1}. Otherwise, i^{th} flower remains at X_i^t.

FPA has gained attention due to its linear nature and its effectiveness in the recent past and large number of improvements in its basic form as been done since its inception.

The revised improved FPA seeks to achieve enhanced performance by enhancing the basic FPA's exploration and exploitation capabilities. Exploration has been improved by the introduction of the Elite Opposition-Based Learning (EOBL) strategy [37], Global pollination phase has been improved by Cauchy based step size that helps to explore the search space more effectively [38], exploitation has been enhanced by Local Neighborhood Search (LNS) driven by knowledge of the best solution so far found in a small neighborhood of the current solution [39]. A balance between the exploration and exploitation is achieved using the dynamic switch probability (p_t) [40], The catfish-effect mechanism [41] is presented to circumvent premature convergence. The major alterations are discussed as follows:

3.1 Strategy for elite opposition-based learning (EOBL)

In basic FPA, once the algorithm falls to the local optimal, it is difficult to achieve the optimal global solution. So, directing the current solution space approximation to the global optimal solution space is very important. The EOBL approach is adopted to improve the FPA's global search capabilities [39].

We would clarify opposition-based learning (OBL) first, before presenting the EOBL. OBL's main idea is that it produces the current solution's opposition solution, simultaneously compares the current solution and the opposition solution, and selects the stronger one to join the next iteration. We assume that $x = (x_1, x_2, x_3, ..., x_D)$ is a point in the population (D is the search space dimension; $x_j \in [a_j, b_j], j = 1, 2, ..., D$) and its opposition point is defined by $\tilde{x} = (\tilde{x}_1, x_2, \tilde{x}_3, ..., \tilde{x}_D)$ as follows:

$$\tilde{x}_j = a_j + b_j - x_j \tag{4}$$

Sometimes the created OBL opposition solution may not be promising than the current search space to search for the optimal global solution, we can therefore use EOBL strategy [39] in this paper. EOBL is an intelligence computing technique. In this, suppose the elite (optimal) individual in the current population is $X_e = (x_{e,1}, x_{e,2}, x_{e,3}, ..., x_{e,D})$. For the individual$X_i$, the elite opposition solution $\tilde{X}_i$ is given by

$$\tilde{x}_{i,j} = \eta * (da_j + db_j) - x_{e,j} \tag{5}$$

where i = 1, 2,..., NP, j = 1, 2,..., D, NP is the population size, and $\eta \in U(0, 1)$ is a generalized coefficient, and $[da_j, db_j]$ is the dynamic boundary of the jth dimensional search space and is represented by:

$$da_j = \min(x_{i,j})$$
$$db_j = \max(x_{i,j}) \tag{6}$$

The dynamic boundary is used instead of fixed boundary in order to preserve the search experience to make the opposite solution situated in the narrowing search space. In addition, if the dynamic boundary operator makes $\tilde{x}_{i,j}$ jump from $[da_j, db_j]$, the following approach is employed to reset $\tilde{x}_{i,j}$:

$$\tilde{x}_{i,j} = rand(da_j, db_j) \tag{7}$$

In EOBL, opposition population is generated according to the elite individual. It makes full use of the elite individuals characteristics to provide more useful search information which will help to enhance the diversity of the population, and hence the global exploration tendency of FPA.

3.2 Cauchy distribution based global pollination phase

Each flower in the global pollination phase of basic FPA updates its position according to the following equation:

$$X_i^{t+1} = X_i^t + \gamma L(\lambda)(X_{best}^t - X_i^t) \tag{8}$$

where X_i^t and X_i^{t+1} are the previous and current positions of i^{th} flower, respectively, X_{best}^t is the flower with best fitness at iteration t, and γ is the scaling factor to control the global pollination phase step size. The function of Lévy flight parameter L is to strengthen the pollination in the conventional FPA, and the step size (λ) obeys the Lévy distribution:

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, (s \gg s_0 > 0) \tag{9}$$

where $\Gamma(\lambda)$ is the gamma function.

In the proposed AFPA, heavy tailed and highly directed Cauchy based step size is utilized instead of Lèvy flights used in conventional FPA [36]. Due to its heavy tailed distribution, this Cauchy based step size is better at exploring the search space and is given by

$$dis = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{\delta}{g}\right) \tag{10}$$

The Cauchy density function is represented as

$$f_{Cauchy(0,g)}(dis) = \frac{1}{\pi}\frac{g}{g^2 + x^2} \tag{11}$$

where g is a scaling parameter with value equals to 1, δ is the Cauchy random operator and dis is a uniform random number. The position updating equation for global pollination phase using Cauchy distribution function is given by:

$$X_i^{t+1} = X_i^t + C(\delta)(X_{best}^t - X_i^t) \tag{12}$$

3.3 Local Neighborhood Search (LNS)

FPA makes use of the best current solution and random solutions to improve local search in the local pollination phase. The local neighborhood search model (LNS) [41] is introduced to further strengthen the local search functionality of basic FPA.

The main idea is to use the current best solution to change the current solution in a small neighborhood of the current solution. For updating the position of the individual, the knowledge of an individual's neighborhood (i.e. the graph of their interconnections called the neighborhood structure) is utilized.

Suppose the population $X = (X_1, X_2, X_3, ..., X_{NP})$, $X_i (i \in [1, NP])$ is a vector in the current population. Here,

the indices of each vector are random in order to maintain the diversity of each neighborhood. Next, we can define the radius r neighborhood (r is a nonzero integer and $2r + 1 < NP$), for each vector X_i, neighborhood of X_i consists of $X_{i-k}, ..., X_i, ..., X_{i+k}$. Figure 1 shows the notion of local neighborhood model in that the vectors can be organized into a ring topology according to their indices.

Mathematically, the LNS model is defined as

$$L_i = X_i + m * \left(X_{n\_opt} - X_i\right) + n * \left(X_p - X_q\right) \tag{13}$$

where $p, q \in [i - r, i + r]$ ($p \neq q \neq i$) and m and n are the scaling factors, where m, $n \in$ rand() and X_{n_opt} is the best solution in the X_i neighborhood. The enhanced version of FPA updates the best solution according to (13), and the modified solution performs the local pollination step as

$$X_i^{t+1} = L_i^t + r * \left(X_k^t - X_m^t\right) \tag{14}$$

where L_i is the best solution updated by LNS and $X_k^t and X_m^t$ are the random solutions of k^{th} and m^{th} flower where $k \neq m$ and r is a scaling factor, where $r \in$ rand().

3.4 Dynamic switch probability

In adaptive FPA, instead of using a fixed switch, an adaptive switch (p_t) has been designed to balance the exploitation and exploration tendency during the search process [42]. The search agents can adapt this strategy to update their next position in accordance with the present fitness value variation, as follows:

$$X_i^{t+1} = \begin{cases} X_i^t + C(\delta)\left(X_{best}^t - X_i^t\right), & rand > p_t \\ X_i^t + m * \left(X_{n\_opt}^t - X_i^t\right) + n * \left(X_p^t - X_q^t\right) & rand \leq p_t \\ \qquad + r * \left(X_k^t - X_m^t\right), \end{cases} \tag{15}$$

here p_t is evaluated in the previous iteration. To accelerate the optimizer convergence, the exploitation must be preferred with a higher probability as compared with the exploration mode. So, the switch p_{t+1} is defined in the range [0.5, 1] with the initial switch value as 0.5, the adaptive transition is given as [36],

$$p_{t+1} = \begin{cases} \dfrac{1}{1 + \exp\left(-\dfrac{f_t^*}{f_{t-1}^*}\right)}, & log_{10}\left|f_t^*\right| \neq log_{10}\left|f_{t-1}^*\right| \\ \dfrac{1}{1 + \exp\left(-\dfrac{f_t^* - \theta.\dfrac{f_t^*}{\theta}}{f_{t-1}^* - \theta.\dfrac{f_{t-1}^*}{\theta}}\right)}, & otherwise \end{cases} \tag{16}$$

In Eq. 16, f_t^* is the finest fitness value obtained at the t^{th} iteration,. is the floor function, θ is the threshold value of adaptive scale parameter that helps to auto-recognize the search state and is given by,

$$\theta = 10^{log_{10}\left|f_t^* - f_{t-1}^*\right| + 1} \tag{17}$$

For the case $f_t^* \ggg f_{t-1}^*$, there is a large difference in fitness values between two iterations. So, the adaptive switch p_{t+1} attains value 1. Therefore, the algorithm switches to the exploitation mode for next iteration. For $f_t^* \lll f_{t-1}^*$, the adaptive switch p_{t+1} will be 0.5 and the exploration mode is selected for next iteration. A local minima has been found for the situation $log_{10}\left|f_t^*\right| = log_{10}\left|f_{t-1}^*\right|$. In order to make this adaptive switch more sensitive, the adaptive switch ratio is modified by the term $\frac{f_t^* - \theta.\frac{f_t^*}{\theta}}{f_{t-1}^* - \theta.\frac{f_{t-1}^*}{\theta}}$. This improvement allows the search agents to jump out of potential traps with higher probability [40].

3.5 Catfish effect mechanism

In real life, fishermen always place catfish into a sardine pond to maintain the freshness of sardines. The catfish disturb the sardines' living environment to activate their ability to survive. This phenomenon derives the catfish effect and was successfully incorporated into PSO [43].

This mechanism is employed to avoid premature convergence by forcing the worst individuals to explore new regions and possibly get better candidate solutions. According to this mechanism if in n consecutive iterations the fitness value of the current best solution has not been enhanced, the 10% worst "sardine" individuals WX will be replaced by new "catfish" individuals CX. The "catfish" individuals are considered as opposition "sardine" individuals, and can be calculated as follows:

$$CX_{id} = a_d + b_d - WX_{id} \tag{18}$$

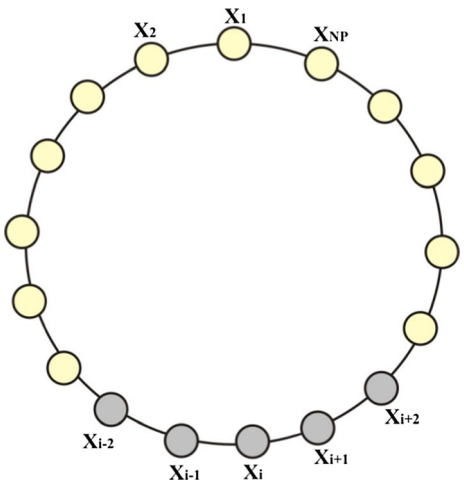


Fig. 1 Neighborhood ring topology of radius 2

where i is the "sardine" individuals index and WX is 10% worst "sardine" individuals.

Main Procedure of the AFPA. The modified approach AFPA is developed by integrating the EOBL technique, Cauchy distribution based Global pollination phase, the LNS model, dynamic switch probability and catfish effect mechanism in the conventional FPA. To demonstrate our proposed algorithm, the detailed pseudocode of the AFPA is shown in Algorithm 1.

Table 1 Parameter settings

Algorithm	Parameters
FPA	NP = 30; I_{max} = 500; p_s = 0.5
GWO	NP = 30; I_{max} = 500; a = [2 to 0]; C = [0 to 2]
SSA	NP = 30; I_{max} = 500;c_1 = [2 to 0]
CS	NP = 30; I_{max} = 500; p_a = 0.25
SCA	NP = 30; I_{max} = 500;r_1 = [2 to 0]
AFPA	NP = 30; I_{max} = 500; p_t = Dynamic and adaptive

Here, NP is the population size, I_{max} is number of iterations

Algorithm 1: Pseudocode of AFPA.

Input: Define objective function $f(x)$, $x = (x1, x2, \ldots, xD)$
Output: Identify current best solution $\boldsymbol{X_{best}^t}$;
Initialization: Initialize related parameters,
Initialize dynamic boundary of the search space,
Randomly initialize a population P of NP random flowers
while stop criterion is not satisfied ***do***
Update the current population with EOBL according to equations (4-7);
while *iterations < maximum number of iterations*
for *i = 1: NP*
if *rand <* $\boldsymbol{p_t}$
perform Global Pollination: $\boldsymbol{X_i^{t+1} = X_i^t + C(\delta)(X_{best}^t - X_i^t)}$
else
perform Local Pollination using equation (13) and (14)
end if
evaluate $\boldsymbol{X_i^{t+1}}$
if $\boldsymbol{X_i^{t+1}}$ *is better than* $\boldsymbol{X_i^t}$, *update*
end if
Update $\boldsymbol{p_t}$ *using equation (16) and (17)*
end for
Find the pollen with the best fitness in the population.
Update $\boldsymbol{X_{best}^t}$ *if the current best pollen beats the previous best pollen.*
Perform ***catfish effect mechanism*** *according to equation (18).*
Return to the next generation until stop criterion is reached.
end while
update final best $\boldsymbol{X_{best}^t}$
end

4 Statistical testing of adaptive FPA algorithm

A set of 10 CEC 2019 benchmark functions (see Appendix Table 5) termed as the "100-Digit Challenge" [38] is selected for AFPA evaluation. Out of these 10 benchmarking functions few of them are shifted and rotated (CEC04-CEC-10), whereas CEC (01–03) are taken as it is. These are all scalable test functions. Comparison with FPA [36], GWO [43], SSA [44], CS [45], and SCA [46] is done to check the effectiveness of AFPA. For every algorithm a total of 500 iterations are performed using 30 agents. The parameter settings for abovesaid algorithms are given in Table 1. For all the algorithms under evaluation, the outcomes are described in terms of the worst, best, mean and standard deviation values for 30 independent runs. AFPA achieves better results than competitive algorithms, except in CEC04, CEC06, and CEC9 as shown in Table 2. Figure 2 displays the convergence graph and box plot of various algorithms for the different benchmark functions.

It has been found that AFPA's results for function CEC01 are better in terms of the worst, average, and median values obtained in compassion to other algorithms, but GWO outperforms in terms of the best fitness performance obtained. GWO, SSA, CS and AFPA are able to find optimal solution for function CEC02, but it is found that AFPA is the better for minimum standard deviation achieved. For the CEC03 function, most of the algorithms can achieve almost globally optimal solution; but overall, in terms of minimum standard deviation, AFPA is found to be better.

For function CEC04, GWO outperforms in terms of the best fitness value achieved, but it has been found that

Table 2 Results of competitive algorithms for CEC 2019 test suite

Function	Parameters	Best	Worst	Average	Median	SD
CEC01	FPA	1.66E + 08	1.34E + 09	4.94E + 08	4.58E + 08	2.36E + 08
	GWO	4.04E + 04	1.22E + 09	1.17E + 08	3.12E + 07	2.39E + 08
	SSA	1.87E + 08	4.03E + 10	7.73E + 09	4.76E + 09	1.04E + 10
	CS	1.00E + 10	1.00E + 10	1.00E + 10	1.00E + 10	0.00E + 00
	SCA	1.60E + 08	2.08E + 10	6.90E + 09	4.44E + 09	6.70E + 09
	AFPA	1.05E + 06	1.75E + 08	2.52E + 07	1.52E + 07	3.59E + 07
CEC02	FPA	1.80E + 01	3.18E + 01	2.14E + 01	2.04E + 01	3.35E + 00
	GWO	1.73E + 01	1.74E + 01	1.73E + 01	1.73E + 01	2.11E-02
	SSA	1.73E + 01	1.73E + 01	1.73E + 01	1.73E + 01	4.57E-04
	CS	1.73E + 01	1.73E + 01	1.73E + 01	1.73E + 01	2.86E-04
	SCA	1.74E + 01	1.77E + 01	1.75E + 01	1.75E + 01	8.77E-02
	AFPA	1.73E + 01	1.73E + 01	1.73E + 01	1.73E + 01	2.80E-05
CEC03	FPA	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01	1.56E-07
	GWO	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01	2.01E-06
	SSA	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01	2.93E-12
	CS	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01	9.81E-11
	SCA	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01	1.00E-04
	AFPA	1.27E + 01	1.27E + 01	1.27E + 01	1.27E + 01	2.78E-14
CEC04	FPA	9.12E + 01	1.98E + 02	1.26E + 02	1.24E + 02	2.54E + 01
	GWO	2.35E + 01	1.43E + 03	1.60E + 02	5.59E + 01	3.41E + 02
	SSA	1.19E + 01	7.86E + 01	3.74E + 01	3.68E + 01	1.85E + 01
	CS	3.67E + 01	5.82E + 02	1.94E + 02	1.78E + 02	1.19E + 02
	SCA	4.94E + 02	2.25E + 03	1.40E + 03	1.31E + 03	4.08E + 02
	AFPA	2.06E + 01	4.09E + 01	2.92E + 01	2.90E + 01	5.41E + 00
CEC05	FPA	1.35E + 00	1.80E + 00	1.60E + 00	1.59E + 00	9.38E-02
	GWO	1.05E + 00	1.85E + 00	1.41E + 00	1.29E + 00	2.72E-01
	SSA	1.04E + 00	1.74E + 00	1.24E + 00	1.19E + 00	1.44E-01
	CS	1.04E + 00	2.00E + 00	1.44E + 00	1.44E + 00	2.50E-01
	SCA	2.06E + 00	2.38E + 00	2.21E + 00	2.19E + 00	7.55E-02
	AFPA	1.03E + 00	1.08E + 00	1.06E + 00	1.06E + 00	1.25E-02
CEC06	FPA	8.94E + 00	1.16E + 01	1.04E + 01	1.05E + 01	7.77E-01
	GWO	7.78E + 00	1.20E + 01	1.06E + 01	1.07E + 01	9.26E-01
	SSA	1.71E + 00	8.55E + 00	4.69E + 00	4.49E + 00	1.76E + 00
	CS	7.73E + 00	1.01E + 01	9.18E + 00	9.25E + 00	6.53E-01
	SCA	9.20E + 00	1.18E + 01	1.08E + 01	1.10E + 01	6.45E-01
	AFPA	8.25E + 00	1.16E + 01	1.03E + 01	1.03E + 01	7.95E-01
CEC07	FPA	2.65E + 01	4.93E + 02	2.84E + 02	2.94E + 02	8.92E + 01
	GWO	-1.34E + 02	1.02E + 03	4.69E + 02	4.30E + 02	3.61E + 02
	SSA	-1.27E + 02	7.61E + 02	2.69E + 02	2.43E + 02	2.24E + 02
	CS	-1.50E + 02	3.63E + 02	1.41E + 02	1.51E + 02	1.17E + 02
	SCA	3.66E + 02	1.05E + 03	7.04E + 02	7.31E + 02	1.70E + 02
	AFPA	-4.56E + 01	1.81E + 02	7.83E + 01	8.56E + 01	6.45E + 01
CEC08	FPA	4.96E + 00	6.24E + 00	5.66E + 00	5.67E + 00	3.27E-01
	GWO	3.39E + 00	6.83E + 00	5.09E + 00	5.14E + 00	9.83E-01
	SSA	2.49E + 00	6.38E + 00	5.05E + 00	5.05E + 00	7.80E-01
	CS	4.59E + 00	5.80E + 00	5.31E + 00	5.28E + 00	3.13E-01
	SCA	4.68E + 00	6.62E + 00	5.96E + 00	6.01E + 00	5.02E-01
	AFPA	2.94E + 00	5.63E + 00	4.59E + 00	4.86E + 00	8.36E-01

Table 2 (continued)

Function	Parameters	Best	Worst	Average	Median	SD
CEC09	FPA	3.70E + 00	6.28E + 00	4.72E + 00	4.58E + 00	6.46E-01
	GWO	2.61E + 00	7.70E + 00	4.52E + 00	4.49E + 00	1.02E + 00
	SSA	2.36E + 00	2.73E + 00	2.49E + 00	2.47E + 00	9.24E-02
	CS	2.56E + 00	3.09E + 00	2.78E + 00	2.76E + 00	1.31E-01
	SCA	1.11E + 01	3.89E + 02	1.09E + 02	9.02E + 01	8.22E + 01
	AFPA	2.42E + 00	4.72E + 00	3.02E + 00	2.96E + 00	5.23E-01
CEC10	FPA	2.02E + 01	2.06E + 01	2.04E + 01	2.04E + 01	9.49E-02
	GWO	2.03E + 01	2.06E + 01	2.05E + 01	2.05E + 01	8.86E-02
	SSA	2.00E + 01	2.03E + 01	2.00E + 01	2.00E + 01	8.87E-02
	CS	2.01E + 01	2.03E + 01	2.03E + 01	2.03E + 01	5.31E-02
	SCA	2.03E + 01	2.07E + 01	2.05E + 01	2.05E + 01	7.70E-02
	AFPA	1.01E + 01	2.06E + 01	1.98E + 01	2.04E + 01	2.35E + 00

AFPA's results are better for the worst, average, and median values attained. AFPA is better than others in the case of CEC05, in terms of all performance metric obtained. SSA's and AFPA's results are better compared with competitive algorithms for function CEC06 and CEC07 functions respectively. AFPA's results for function CEC08 are better for the worst, average, and median values obtained, however, the results of SSA achieved best fitness function value. AFPA's results are competitive with other algorithms for function CEC09 and CEC10.

Figure 2 displays the box-plots of competitive algorithms for CEC 2019 benchmark functions. The box-plots here are used to measure algorithm efficiency in terms of fitness values. It can be shown that in most cases the proposed AFPA is cost-effective as median of fitness values for AFPA is lower. Therefore, AFPA's overall performance is found to be consistent with other algorithms for optimization.

5 Location determination using single anchor node

The concept of deploying a single AN in the sensor field is used to locate target nodes. These nodes are deployed into three different layers, anchors are placed at the top, whereas at the bottom and middle layer TNs are deployed randomly. These ANs transmit a beacon signal that helps the TNs to locate themselves. As TNs fall within the range of AN, then by using the beacon signal, TNs obtain the RSSI information of an anchor and then calculating Euclidean distance between AN and TN. After calculating Euclidean distance six VAs are projected at 60 degrees each using umbrella projection concept to locate TNs, as shown in Fig. 3. For locating TNs in 3D scenario, minimum 4 anchors are required (as shown in Fig. 4), one anchor and three VAs are used to obtain the centroid. Then using this centroid information, Adaptive FPA algorithm is applied for optimization.

By using this concept of VAs, LoS issues are also minimized to a greater extent.

$$d_i = \sqrt{(x_t - x)^2 + (y_t - y)^2 + (z_t - z)^2} \tag{19}$$

where $(x_t,\ y_t, z_t)$ represents the TN position and (x,y,z) represents current location of the AN.

The centroid position $(x_c,\ y_c, z_c)$ is calculated using AN and the respective virtual ANs $(xv_1,\ yv_1, zv_1)$, $(\ xv_2,\ yv_2, zv_2), and (xv_3, yv_3, zv_3)$ as shown in Fig. 5.

The centroid coordinates are determined as

$$(x_{c,}y_c, z_{c,}) = \left(\frac{x + xv_1 + xv_2 + xv_3}{3}, \frac{y + yv_1 + yv_2 + yv_3}{3}, \frac{z + zv_1 + zv_2 + zv_3}{3}\right) \tag{20}$$

After obtaining the centroid, random particles are deployed as shown in Fig. 6. Adaptive FPA is used to locate all the moving TNs and to estimate the coordinates as $x_{s,}y_s, z_{s,}$. The fitness function is obtained by calculating mean square error between actual and estimated coordinates distances of actual and estimated nodes as given in (21) is minimized.

$$f(x_{s,}y_{s,}z_{s,}) = \frac{1}{M}\sum \left(\sqrt{(x_e - x_i)^2 + (y_e - y_i)^2 + (z_e - z_i)^2} - d_i^{\wedge} \right)^2 \tag{21}$$

Here x_i, y_i, z_i represents the location of placed beacon node and x_e, y_e, z_e represents the estimated location of TNs (M > 4 to compute 3D location).

Other meta-heuristic algorithms perform in the similar fashion to localize themselves.

Each algorithm obtains the optimum location as shown in Fig. 7, and localization error is calculated as given in (22)

$$E_t = \frac{1}{N_L} \sum \sqrt{(x_e - x_t)^2 + (y_e - y_t)^2 + (z_e - z_i)^2} \qquad (22)$$

where N_L represents localized TNs.

6 Results and discussions

In this research work, a novel approach of using a single anchor with their projection in hexagonal directions to obtain 3D locations of TNs using various meta-heuristic optimization algorithms. Using MATLAB software, various meta-heuristic algorithms are tested for 80 TNs deployed at middle and bottom

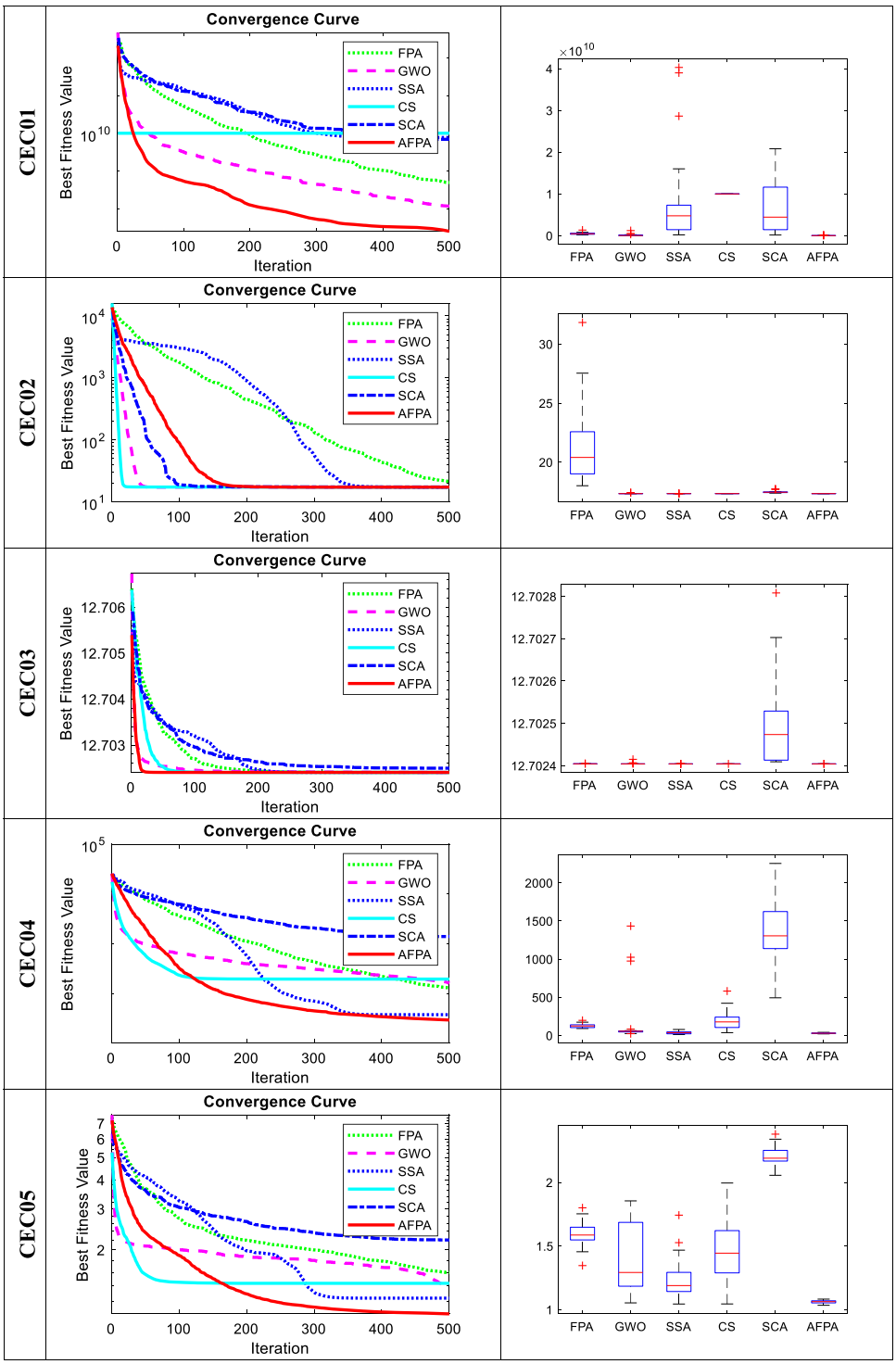
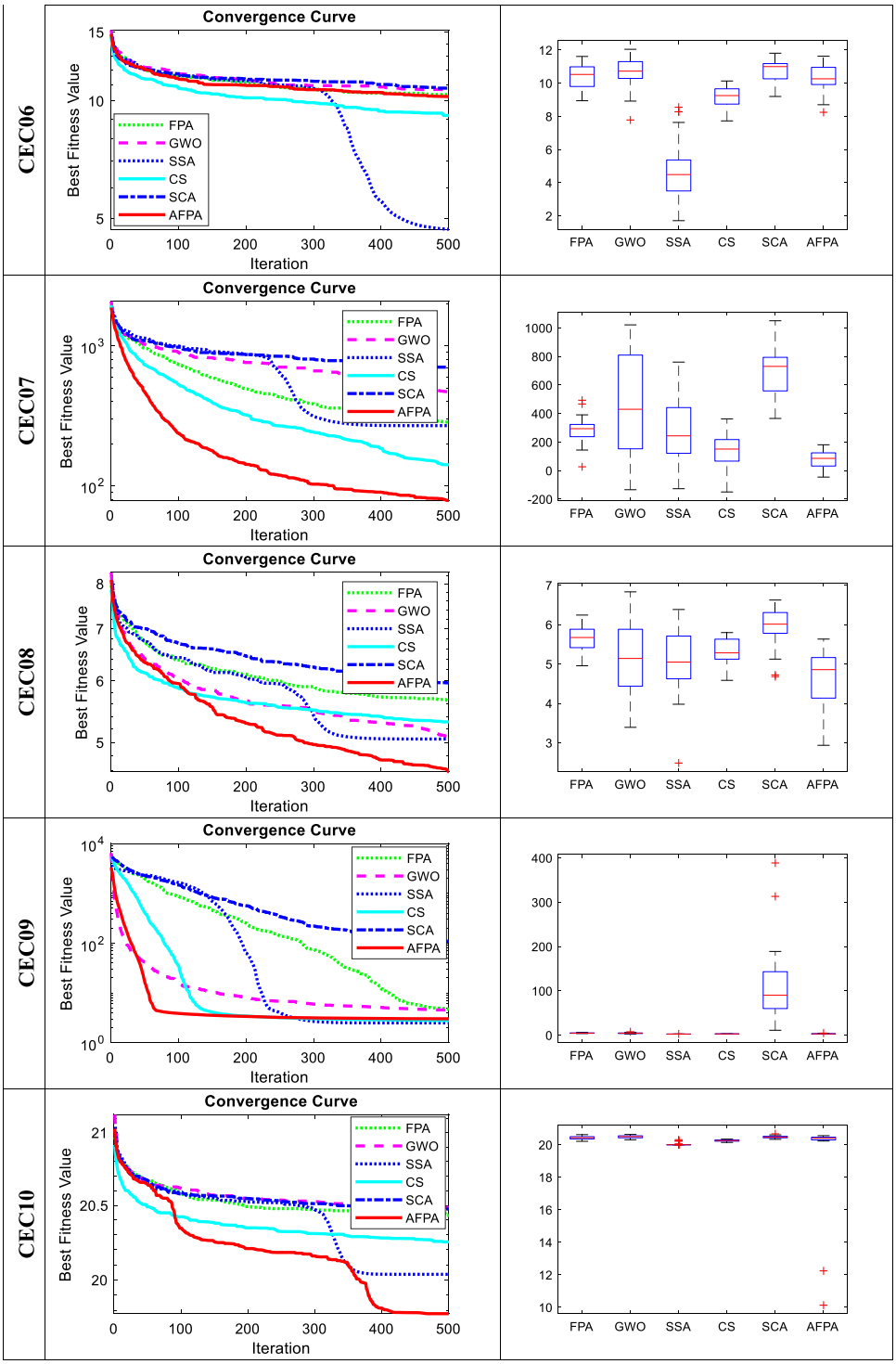


Fig. 2 Convergence graph and boxplot for CEC 2019 test suite

Fig. 2 continued



layer (40 each). The results were performed in WSN field with single AN on the MacBook Air i5 with 4 GB RAM processing. A cubic structure with different properties of TNs and three-layer structure is considered. In this, three-layer arrangement at the top most layer a single AN and at bottom and middle layer 40 TNs are deployed randomly. For obtaining 3D coordinates at least four anchors (one anchor and minimum three VAs) are required. Whenever, the moving TNs fall within the range of AN, the above said scenario is considered. An umbrella projection is formed with the help of anchor and VAs is formed for determining the 3D positions of target nodes.

Various meta heuristic algorithms are applied in this research for obtaining 3D coordinate. Different parameters chosen for meta-heuristic algorithms are as shown in Table 3.

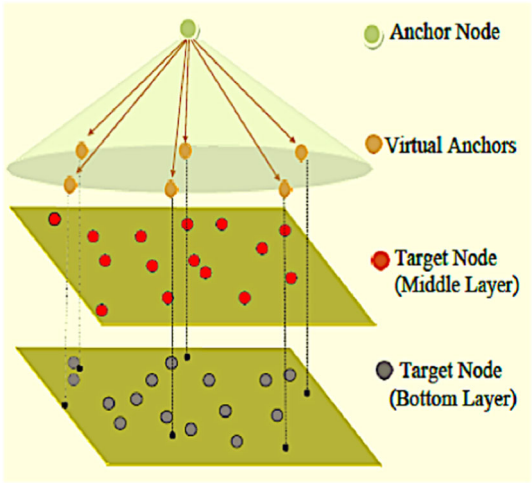


Fig. 3 Umbrella projection for deploying anchor and virtual anchors

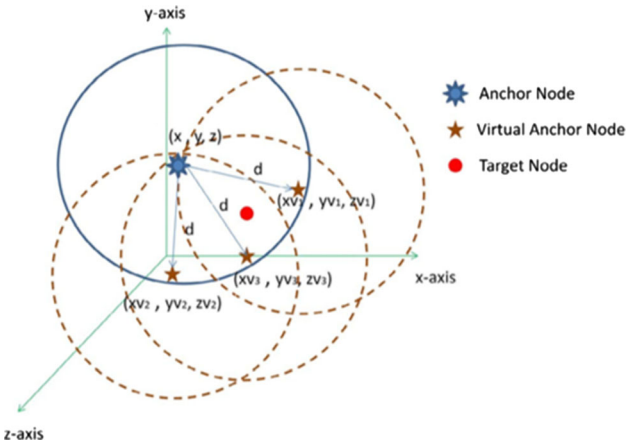


Fig. 4 Anchor and projection of VA in 3D environment

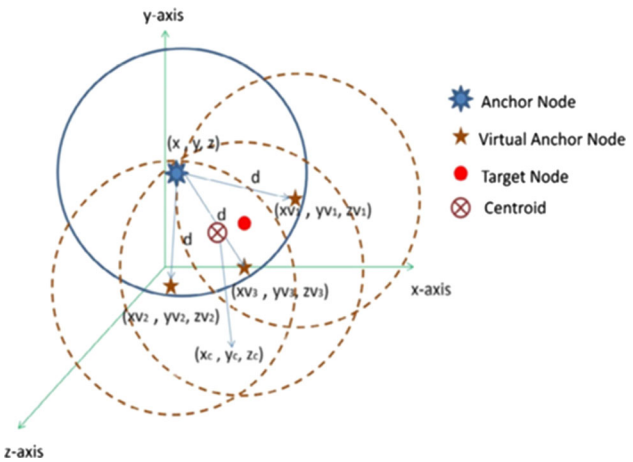


Fig. 5 Calculating centroid in 3D scenario

In mobility based scenario, various meta-heuristic algorithms are evaluated in the following section. The TNs are deployed initially in a random fashion at bottom and

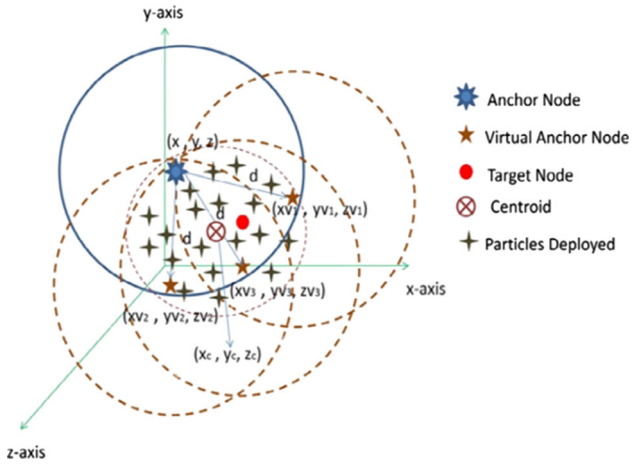


Fig. 6 Adaptive FPA deployed around centroid

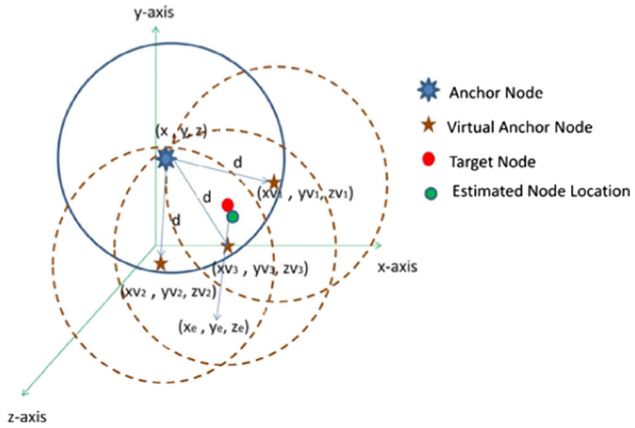


Fig. 7 Target estimation using adaptive FPA

middle layers and at the top most layer is equipped with AN. AN is fixed while mobility is applied for TNs. The fitness function is calculated using the average of localization error. The results of various meta-heuristic algorithms are shown in Figs. 8, 9, 10, 11, 12, 13, 14, 15, 16. From these results, it is clear that the anchor forms umbrella projection pattern with VAs and TN to determine the 3D positions. By using VA concept, the problem of LoS is minimized to greater extent. From these observations it is quite clear that Adaptive FPA outperforms in context of other meta-heuristic optimization algorithms and have faster convergence rate.

In static scenario, a smaller range and only few anchors are considered. All TNs get localized using this AN and referred as pseudo anchor once it is localized. Then pseudo anchors and initially deployed anchors collectively work together to locate the other unlocated TNs. In static deployment the convergence rate is slow and requires a greater number of rounds to locate all TNs. Whereas in dynamic scenarios, AN with higher range is considered and more number of TNs are located in shorter time.

Table 3 Parameter settings

Algorithm	Parameters
PSO	NP = 30; D = 3; I_{max} = 100; c_1, c_2, c_3 = 1.494; w = 0.729
HPSO	NP = 30; D = 3; I_{max} = 100; c_1, c_2, c_3 = 1.494; η = 0.1; w = 0.729
BBO	NP = 30; D = 3; I_{max} = 100; p_m = 0.05
FA	NP = 30; D = 3; I_{max} = 100;$\alpha = 0.2; \gamma = 0.96$
GWO	NP = 30; D = 3; I_{max} = 100; a = [2 to 0]; C = [0 to 2]
NMRA	NP = 30; D = 3; I_{max} = 100; $\alpha = 0.5; \gamma = 1$; w = adaptive;$\beta_0 = 1; \beta_{min} = 0.2$
NMRV 3.0	NP = 30; D = 3; I_{max} = 100;$\lambda = U(0,1)$
FPA	NP = 30; D = 3; I_{max} = 100; p_s = 0.5
Adaptive FPA	NP = 30; D = 3; I_{max} = 100; p_t = Dynamic and adaptive

Here, NP is the population size, D is dimension of problem, I_{max} is number of iterations

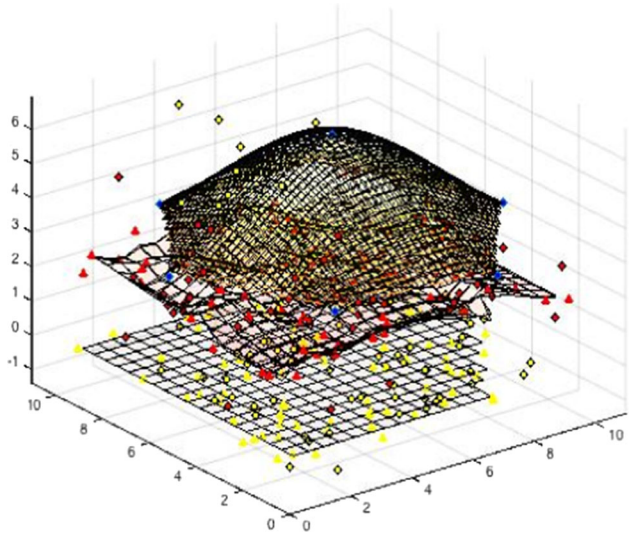


Fig. 8 Target node movement using BBO

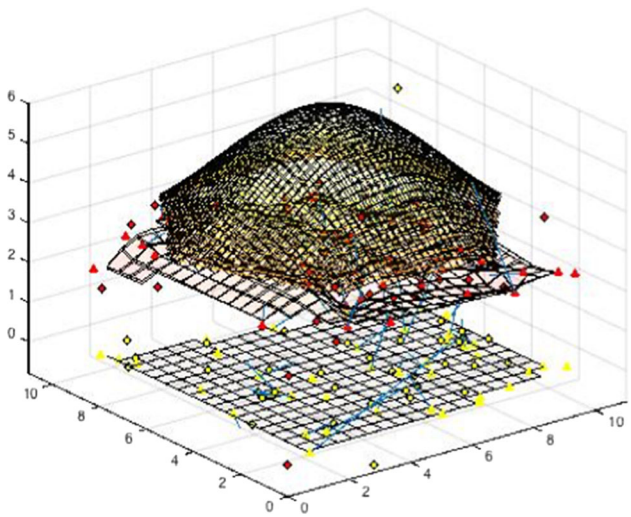


Fig. 9 Target node movement with HPSO

Initially, anchors and TNs are deployed in similar fashion for all meta-heuristic algorithms. To compare the

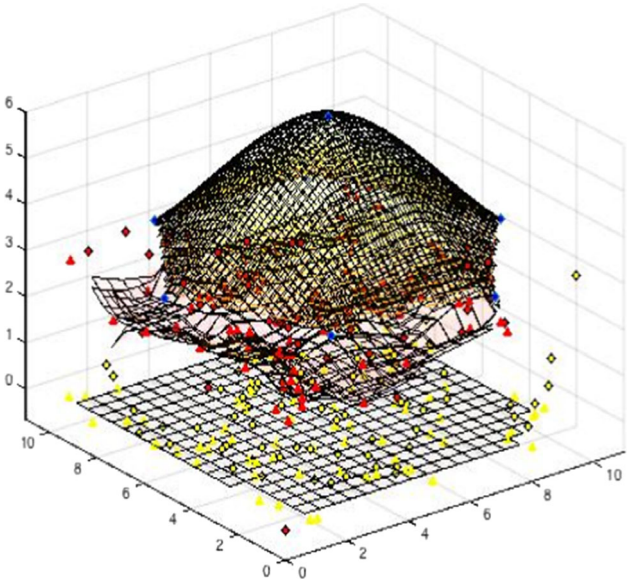


Fig.10 Target node movement with FA

results in mobility scenario various performance metrics are considered as described in Table 4. The convergence time of Adaptive FPA is quite less as compared to other competitive algorithms tested for the same scenario.

7 Conclusions and future scope

Range based and range free localization techniques are the popular available techniques to locate unknown nodes. In this research, range-based techniques along with various meta-heuristic algorithms are used to obtain 3D coordinates of TNs using single AN concept. Anchor node and the VAs form an umbrella projection for locating all TNs. Whenever the moving targets fall within the scope of AN, three VAs and anchor itself works unitedly to locate the TNs (as minimum 4 nodes are required to obtain 3D positions). In this approach, node is estimated with least

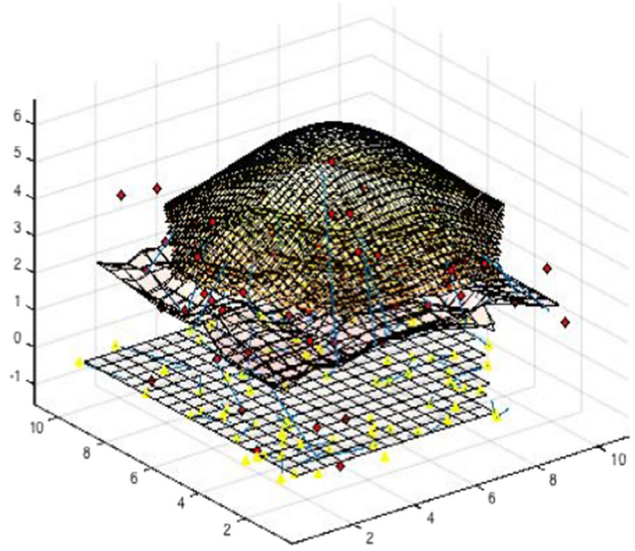


Fig.11 Target node movement with HPSO

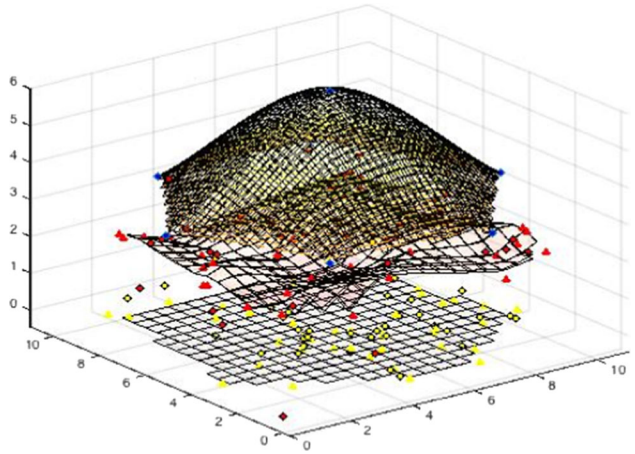


Fig.12 Target node movement with GWO

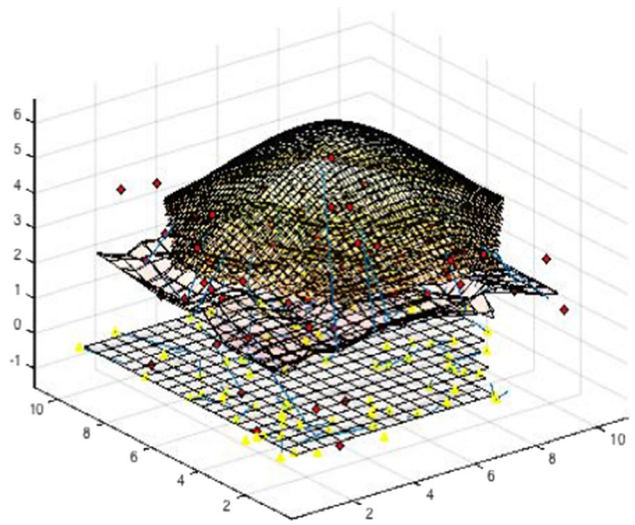


Fig.13 Target node movement with FPA

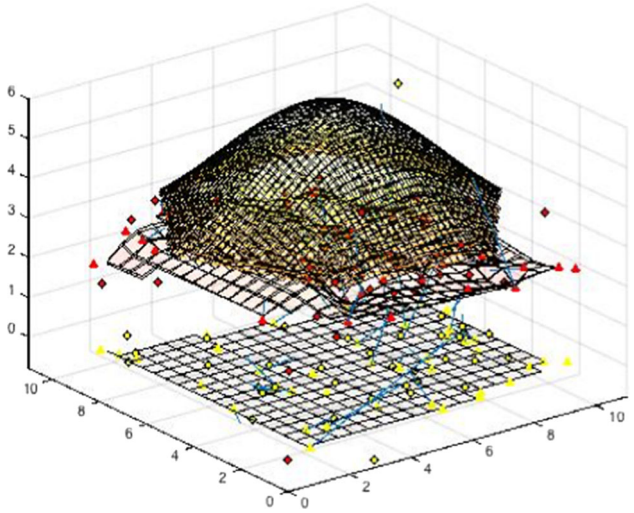


Fig.14 Target node movement with NMRA

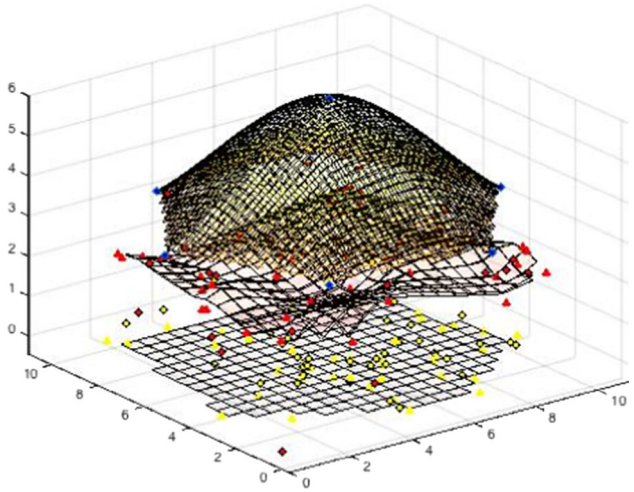


Fig.15 Target node movement with NMRV 3.0

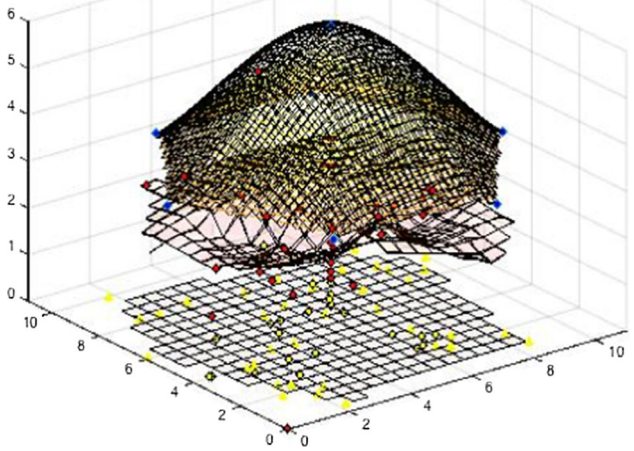


Fig.16 Target node movement with adaptive FPA

Table 4 Comparison of meta-heuristic algorithms

Optimization	Number of movements	Localization error (max)	Localization error (min)	Average error	Number of located targets
PSO	1	3.9358	0.0554	0.9958	80
	2	5.3379	0.0831	0.9839	80
	3	5.0108	0.0800	0.9267	80
	4	5.1655	0.0367	0.9757	80
	5	5.1325	0.0812	0.9612	80
HPSO	1	3.1204	0.1044	0.6742	80
	2	5.0134	0.0647	0.4876	80
	3	4.8279	0.0976	0.4032	80
	4	5.2376	0.0230	0.5546	80
	5	5.2134	0.0316	0.5324	80
BBO	1	5.8904	0.1822	1.1892	80
	2	5.3500	0.3318	1.2560	80
	3	5.5989	0.1822	1.1585	80
	4	5.6348	0.1528	1.2818	80
	5	5.9014	0.1911	1.1916	80
FA	1	6.1101	0.1922	2.2234	80
	2	6.3120	0.3412	2.3124	80
	3	6.6990	0.1923	2.4651	80
	4	6.8912	0.1627	2.5123	80
	5	6.9036	0.2010	2.2013	80
GWO	1	3.1101	0.0944	0.6442	80
	2	4.9834	0.0547	0.4776	80
	3	4.8134	0.0876	0.3932	80
	4	4.7976	0.0430	0.4946	80
	5	4.9776	0.0513	0.4713	80
NMRA	1	4.686	0.0989	0.6949	80
	2	5.7366	0.1544	0.4928	80
	3	4.7559	0.0994	0.4621	80
	4	5.6522	0.1581	0.5498	80
	5	4.5094	0.0989	0.4923	80
NMRV 3.0	1	3.1101	0.0944	0.6442	80
	2	4.9834	0.0547	0.4776	80
	3	4.8134	0.0876	0.3932	80
	4	4.7976	0.0430	0.4946	80
	5	4.9776	0.0513	0.4713	80
FPA	1	3.1304	0.1044	0.6712	80
	2	5.0234	0.0637	0.4866	80
	3	4.8179	0.0966	0.4022	80
	4	5.2476	0.0220	0.5536	80
	5	5.2234	0.0336	0.5314	80
Adaptive FPA	1	3.1101	0.0964	0.4315	80
	2	4.3983	0.0437	0.4132	80
	3	4.8032	0.0721	0.2741	80
	4	4.7679	0.0412	0.2671	80
	5	4.3108	0.0403	0.2512	80

localization error along with lesser computational time in Adaptive FPA. The computation time for PSO and HPSO is of the order of one fourth of the second, but for BBO and FA, the convergence rates are higher (close to a second)

respectively whereas for Adaptive FPA the convergence time is only 0.1234 s.

The implemented algorithm can be used in a variety of applications, such as animal tracking, logistics, monitoring of coal mine workers and industrial applications. The limitation of this work is that the algorithm is relying only upon a single AN for locating TNs in 3D environment and due to power issues if a single AN gets non-operational, the network gets terminated. So, there is trade-off using the concept of single AN, as only a single AN is required to locate TNs in 3D environment which is cost effective.

Appendix

See Table 5.

Table 5 The 100-digit challenge basic test functions [38]

No	Functions	Fi* = Fi(x*)	D	Search range
1	Storn's Chebyshev polynomial fitting problem	1	9	[− 8192, 8192]
2	Inverse hilbert matrix problem	1	16	[− 16,384, 16,384]
3	Lennard–Jones minimum energy cluster	1	18	[− 4, 4]
4	Rastrigin's function	1	10	[− 100, 100]
5	Griewangk's function	1	10	[− 100, 100]
6	Weierstrass function	1	10	[− 100, 100]
7	Modified Schwefel's function	1	10	[− 100, 100]
8	Expanded Schaffer's F6 function	1	10	[− 100, 100]
9	Happy cat function	1	10	[− 100, 100]
10	Ackley function	1	10	[− 100, 100]

References

1. Karl, H., & Willig, A. (2007). *Protocols and architectures for wireless sensor networks*. New York: Wiley.
2. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine, 40*(8), 102–114.
3. Zhou, G., He, T., Krishnamurthy, S., & Stankovic, J. A. (2006). Models and solutions for radio ir- regularity in wireless sensor networks. *ACM Transaction Sensor Networks (TOSN), 2*(2), 221–262.
4. Robles, J. J. (2014). Indoor localization based on wireless sensor networks. *AEU-International Journal of Electronics and Communication, 68*(7), 578–580.
5. Boukerche, A., Oliveira, H. A., Nakamura, E. F., & Loureiro, A. A. (2007). Localization systems for wireless sensor networks. *IEEE Wireless Communication, 14*(6), 6–12.
6. Cho, H., & Kwon, Y. (2016). Rss-based indoor localization with pdr location tracking for wireless sensor networks. *AEU- International Journal of Electronics and Communication, 70*(3), 250–256.
7. Awad, A., Frunzke, T., & Dressler, F. (2007). Adaptive distance estimation and localization in wsn using rssi measures. In *10th euromicro conference on digital system design ar- chitectures, methods and tools*, DSD 2007. IEEE; 2007. pp. 471–8.
8. Bulusu, N., Heidemann, J., & Estrin, D. (2000). Gps-less low-cost outdoor localization for very small devices. *IEEE Personal Communicaion, 7*(5), 28–34.
9. Graefenstein, J., Albert, A., Biber, P., & Schilling, A. (2009). Wireless node localization based on rssi using a rotating antenna on a mobile robot. In *6th workshop on positioning, navigation and communication*, 2009. WPNC 2009. IEEE; 2009. pp. 253–9.
10. Sumathi, R., Srinivasan, R. (2011) RSS-based location estimation in mobility assisted wireless sensor networks. In *IEEE 6th international conference on intelligent data acquisi- tion and advanced computing systems (IDAACS)*, 2011 (vol. 2, pp. 848–52). IEEE.
11. Guo, Z., Guo, Y., Hong, F., Jin, Z., He, Y., Feng, Y., et al. (2010). Perpendicular intersection: lo- cating wireless sensors with mobile beacon. *IEEE Transactions on Vehicular Technology, 59*(7), 3501–3509.
12. Shi, Q., Huo, H., Fang, T., & Li, D. (2009). A 3d node localization scheme for wireless sensor networks *6*(3), 167–172.
13. Wang, L., Zhang, J., & Cao, D. (2012). A new 3-dimensional dv-hop localization algorithm. *Journal of Computer Information Systems, 8*(6), 2463–2475.
14. Yan, X., Zhuang, Y., & Jing-jing. , G. (2015). An improved 3D localization algorithm for the wireless sensor network. *International Journal of Distributed Sensor Networks*. https://doi.org/10.1155/2015/315714.
15. Li, J., Zhong, X., & Lu, I.-T. (2014). Three-dimensional node localization algorithm for wsn based on differential rss irregular transmission model. *Journal of Communication, 9*(5), 391–397.
16. Ahmad, T., Li, X. J., & Seet, B.-C. (2017). Parametric loop division for 3d localization in wireless sensor networks. *Sensors, 17*(7), 1697.
17. Gopakumar, A., & Jacob, L. Localization in wireless sensor networks using particle swarm optimization.
18. Chuang, P.-J., & Wu, C.-P. An effective pso-based node localization scheme for wireless sensor networks. In *Ninth international conference on parallel and distributed computing, applications and technologies*, 2008. PDCAT 2008. IEEE; 2008. pp. 187–94.
19. Kumar, A., Khosla, A., Saini, J.S., & Singh, S. (2012). Meta-heuristic range-based node localization algorithm for wireless sensor networks. In *International conference on localization and GNSS (ICL-GNSS)*. IEEE; 2012. p. 1–7.
20. Sujatha, S., & Siddappa, M. (2017). Node localization method for wireless sensor networks based on hybrid optimization of particle swarm optimization and differential evo-lution. *IOSR Journal of Computing Engineering, 19*(2), 07–12.
21. Arora, S., & Singh, S. Node localization in wireless sensor networks using butterfly optimization algorithm. *Arabian Journal for Science and Engineering* 2017:1–11.

22. Mautz, R., Ochieng, W., Brodin, G., & Kemp, A. H. (2007). 3d wireless network localization from inconsistent distance observations. *Ad Hoc Sensor Wireless Networks, 3*(2–3), 141–170.
23. Liu, W., Dong, E., & Song, Y. (2016). Analysis of flip ambiguity for robust three-dimensional node localization in wireless sensor networks. *Journal of Parallel Distribution Computing, 97,* 57–68.
24. Bai, S., & Qi, H. (2016). Tackling the flip ambiguity in wireless sensor network localization and beyond. *Digitial Signal Processing, 55,* 85–97.
25. Han, S., Yue, J., Meng, W., & Li, C. (2015). A novel flip ambiguities detection algorithm for wsn localization. In *IEEE globecom workshops*. IEEE; 2015. pp. 1–6.
26. Singh, P., Khosla, A., Kumar, A., & Khosla, M. (2017). A novel approach for localization of moving target nodes in wireless sensor networks. *International Journal of Grid and Distributed Computing, 10*(10), 33–44.
27. Singh, P., Khosla, A., Kumar, A., & Khosla, M. (2018). Computational intelligence-based localization of moving target nodes using single anchor node in wireless sensor networks. *Telecommunication Systems, 69*(3), 397–411.
28. Singh, P., Khosla, A., Kumar, A., & Khosla, M. (2017). 3D localization of moving target nodes using single anchor node in anisotropic wireless sensor networks. *AEU-International Journal of Electronics and Communications, 82,* 543–552.
29. Singh, P., Khosla, A., Kumar, A., & Khosla, M. (2018). Optimized localization of target nodes using single mobile anchor node in wireless sensor network. *AEU-International Journal of Electronics and Communications, 91,* 55–65.
30. Yang, B., Guo, L., Guo, R., Zhao, M., & Zhao, T. (2020). A novel trilateration algorithm for RSSI-based indoor localization. *IEEE Sensors Journal, 20*(14), 8164–8172.
31. Goyat, R., Rai, M. K., Kumar, G., Kim, H.-J., & Lim, S.-J. (2020). Improved DV-Hop localization scheme for randomly deployed WSNs. *International Journal of Sensors Wireless Communications and Control, 10*(1), 94–109.
32. Nguyen, T. L. N., Vy, T. D., & Shin, Y. (2019). An efficient hybrid RSS-AOA localization for 3d wireless sensor networks. *Sensors, 19*(9), 2121.
33. Singh, S. P., & Sharma, S. C. (2019). Implementation of a PSO based improved localization algorithm for wireless sensor networks. *IETE Journal of Research, 65*(4), 502–514.
34. Singh, P., & Mittal, N. (2020) An efficient localization approach for WSNs using hybrid DA-FA algorithm. IET-Communications.
35. Singh, P., & Mittal, N. (2020) Naked mole-rat algorithm with improved exploration and exploitation capabilities to determine 2D and 3D coordinates of sensor nodes in WSNs. AJSE.
36. Yang, X. S. (2012). Flower pollination algorithm for global optimization. In *International conference on unconventional computing and natural computation* (pp. 240–249). Berlin: Springer.
37. Xiuli, W., Yongquan, Z., & Yuting, L. (2017) Elite opposition-based water wave optimization algorithm for global optimization. In *Mathematical problems in engineering,* Article ID 3498363.
38. Singh, U., & Salgotra, R. (2017). Pattern synthesis of linear antenna arrays using enhanced flower pollination algorithm. International Journal of Antennas and Propagation, pp. 1–11.
39. Das, S., Abraham, A., Chakraborty, U. K., & Konar, A. (2009). Differential evolution using a neighborhood-based mutation operator. *IEEE Transactions on Evolutionary Computation, 13*(3), 526–553.
40. Wu, J., Wang, Y. G., Burrage, K., Tian, Y. C., Lawson, B., & Ding, Z. (2020). An improved firefly algorithm for global continuous optimization problems. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2020.113340.
41. Chuang, L. Y., Tsai, S. W., & Yang, C. H. (2011). Chaotic catfish particle swarm optimization for solving global numerical optimization problems. *Applied Mathematics and Computation, 217*(16), 6900–6916.
42. Janez, B., Mirjam, M., Borko, S. B. (2019). The 100-digit challenge: algorithm. pp. 19–26.
43. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69,* 46–61.
44. Mirjalili, S., Gandomi, A. H., Zahra, S., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114,* 163–191.
45. Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. In *2009 World congress on nature & biologically inspired computing (NaBIC)* (pp. 210–214). IEEE.
46. Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems, 96,* 120–133.

Prabhjot Singh received his B.Tech. in Electronics and Communication Engineering (ECE) from Punjab Technical University, Jalandhar, and M.Tech. degree in ECE from Lovely Professional University, Jalandhar, India in 2010 and 2012 respectively. He is pursuing Ph.D. in ECE from Chandigarh University, Mohali. He is working as Assistant Professor with Chandigarh University, Mohali, India. His research interests include Wireless Sensor Networks.

Nitin Mittal received his B.Tech. and M.Tech. degree in Electronics and Communication Engineering (ECE) from Kurukshetra University, Kurukshetra, India in 2006 and 2009 respectively. He has completed his Ph.D. in ECE from Chandigarh University, Mohali, India in 2017. He is presently working as Associate Professor in ECE Department at Chandigarh University, Mohali, India. His research interests include Wireless Sensor Networks, Image Segmentation and Soft Computing.