

Lifetime Improvement in WSN using Flower Pollination Meta Heuristic Algorithm Based Localization Approach

T. Shankar, Tony James, R. Mageshvaran and A. Rajesh

VIT University, Vellore, India; tshankar77@gmail.com, tonyjames.p@gmail.com,
rmageshvaran@vit.ac.in, arajesh@vit.ac.in

Abstract

Objectives: The nodes in a wireless sensor network are extremely energy limited. Often, there occurs transmission of similar sensed data from adjacent nodes of a geographical region with high sensor node density. Network lifetime of the sensor network can be improved by grouping such closely packed nodes using the node locations obtained from a localization algorithm, thereby eliminating redundant data transmission. **Methods:** This paper employs the Flower Pollination Algorithm (FPA) for node localization and the performance of this method is compared against alternative localization techniques, viz. Particle Swarm Optimization (PSO) and Harmony Search Algorithm (HSA). **Findings:** The lifetime improvement of the proposed Proximity Grouping Algorithm (PGA) method is compared with that offered by conventional *Low-Energy Adaptive Clustering Hierarchy* (LEACH) method. **Application/Improvements:** The proposed PGA method shows 35% better network lifetime improvement, based on the residual energy of the network.

Keywords: Flower Pollination Algorithm (FPA), Flower Pollination, Localization, Meta Heuristic, Network Lifetime.

1. Introduction

Over the last decade, with the dominance of wireless technology, challenges and issues faced by wireless sensors have become hot research areas^{1,2}. Data gathering techniques, network lifetime extension, localization of nodes, optimizing network coverage, object tracking are the prominent among these issues. The application of WSNs to different monitoring tasks like search and rescue and habitat monitoring makes the finding the locations of the sensor nodes a vital part of the process³.

The sensor nodes in a wireless sensor network are usually deployed randomly over a big terrain. The positions where the nodes fall cannot be determined beforehand. Oftentimes, some geographical areas might have a high sensor node density as compared to other areas. If the sensor nodes are tasked with sensing geographically relevant data like temperature or humidity, nodes in a high density area are found to transmit similar data⁴. Transmission of data uses up the energy of the individual

nodes and reduce the overall lifetime of the network. It becomes desirable, therefore, to minimize or eliminate this redundant transmission of similar data from nodes in high density areas, so as to improve the lifetime of the sensor network. Localization, or finding the locations of the nodes in the sensor network, plays a vital role in determining the node densities of different geographical areas. Moreover, the data from these nodes will become meaningful only if the location of the nodes of the source data is known as well⁵. The simplest solution to finding locations of the nodes is by installing GPS units in each node. This solution, in addition to being a very costly process, also adds to the hardware requirements of the node^{6,7}. As the nodes are extremely energy-constrained due to their limited batteries, this additional burden renders this solution ineffective. The commonly used alternative approach is to equip only a few nodes with GPS capability⁸, and use the data from these nodes to find the locations of the

*Author for correspondence

nodes of the entire network using a suitable algorithm. These nodes are known as anchor nodes⁹.

This paper proposes the use of a nature-inspired Flower Pollination Meta heuristic algorithm for process of node localization¹⁰. The performance of this method is compared with the performance of other alternative meta heuristic approaches to the node localization problem, viz. Particle Swarm Optimization (PSO)¹¹ and Harmony Search Algorithm (HSA)¹².

The structure of the paper is organized as follows. Section 2 includes the problem formulation. Section 3 presents the proposed Flower Pollination Algorithm (FPA) algorithm and its use for localization. Section 4 describes the working and implementation of alternative Meta heuristic approaches to node localization¹³. Section 5 details the methodology of the proposed Proximity Grouping Algorithm (PGA) method for network lifetime improvement. Section 6 presents the results of simulation and observations. Section 7 concludes the main contributions of this paper.

2. Problem Statement

A wireless sensor field, represented by a regular square region, is considered as the terrain over with r number of sensor nodes are randomly deployed i.e. we cannot predict where the nodes would be placed. Of these r nodes, a few of the nodes, say 's' numbers, are to be taken as anchor nodes (this is usually taken as around 10% of the total number of sensor nodes)¹⁴. The anchor nodes are provided with GPS capability to allow for determining the locations of these few nodes with high precision. Using the location data information of only the few anchor nodes (which are obtained from GPS system), the goal then is to determine the positions of the remaining ($r-s$) sensor nodes. As this is an optimization problem, the minimum value of the objective function that has to be obtained, in this case, is the square of the range errors between non-anchor nodes and the anchor nodes. This can be represented mathematically using the Equation (1).

$$CF_{FPA} = \sum_{k=1}^{N_0} \frac{1}{N_0} \sum_{n=1}^s \left(\sqrt{(x_k - x_n)^2 + (y_k - y_n)^2} - \hat{d}_n \right)^2 \quad (1)$$

where (x, y) are the target node positions represented in the x - y domain, (x_k, y_k) represent the position of the k^{th} anchor node, $\hat{d}_n$ gives a measure of the distance between a

node (non-anchor) and its neighboring anchor nodes. In order to determine how closely the algorithm estimated the location of the sensor nodes, we calculate the node localization error given in Equation (2).

$$NLE = \sqrt{\frac{1}{N_0} \sum_{k=s+1}^{N_0} \|p_k - \hat{p}_k\|^2} \quad (2)$$

Where p_k is the actual position and $\hat{p}_k$ is the estimated location of the k^{th} sensor node and N_0 is the total number of sensor nodes.

3. Flower Pollination Algorithm

3.1 Flower Pollination

Plant reproduction happens when there occurs a union of male and female gametes, a process which is termed pollination¹⁵. In this process, the pollen grains (male gametes) are delivered to the stigma (female part of the flower) where union takes places with female gametes. The result of this union is fertilization and the ovules grow and develop into seeds contained within fruits¹⁶. This action of fertilization is performed with the help of a biotic factor like water and wind, or from biotic factors like insects. Pollination manifests in two forms: self-pollination and cross-pollination. Self-pollination is brought about with pollen grains of a flower is carried within and fertilizes the same flower. Such a phenomenon is observed in flowers which possess both male and female gametes, such as peach flowers, where there is an unavailability of suitable pollinators. Cross-pollination, on the other hand, is brought about when the male gametes (pollen grains) get transported to a different flower of a different plant. The pollen grains, being sticky in nature, are suitably designed to attach to the body of the insects visiting the flower looking for nectar¹⁷. When these insects fly off and visit other flowers, the pollen grains are thereby transported and delivered to the stigma of other flowers.

3.2 Flower Pollination Algorithm

The flower pollination algorithm was developed by Xin-She Yang in 2012. The pollination of flowers could very well be regarded as a process of optimization, which brings about the survival of the fittest of a particular plant species¹⁸. It is this very aspect that is utilized for the sensor node localization. The pseudo code of the proposed algorithm is as follows:

Pseudo code for Proposed FPA for node localization

```

Let  $f(x)$ ,  $x = (x_1, x_2, \dots, x_d)$  be objective function to be minimized
N flowers or gametes in population, given random initial solutions
Fitness of each particle in initial population evaluated and best solution identified (global best)
Switching probability  $p$ , for each flower, chosen randomly,  $p \in [0, 1]$ 
when ( $i < \text{Max Itr}$ )
  for  $k = 1 : N$ 
    Levy step  $L$  selected randomly from Levy distribution
    global pollination done by  $P_k^{i+1} = P_k^i + \gamma L(G^* - P_k^i)$ 
    Check if simple bounds/limits are satisfied
  else
    random  $\epsilon$  taken from a uniform distribution  $[0, 1]$ 
    local pollination done by  $P_k^{i+1} = P_k^i + \epsilon (P_k^i - P_l^i)$ 
    check if simple bounds/limits are satisfied
  end
  Compute new_soln
  if (new_soln > existing)
    update population
  else
    exit;
  end
end
solution_best = present best solution;
end
solution_best

```

The algorithm is dictated by the following set of rules:

1. A biotic and self-pollination brings about local pollination, whereas biotic and cross pollination accomplishes global pollination.
2. When travelling, the steps taken by pollinators tend to obey Levy distribution.
3. Certain insects exhibit loyalty to certain kinds of flowers.
4. The extent of global and local pollination can be controlled by using a switching probability (p).

Pollination of flowers can, in theory, occur at both global and local settings. However, when considering practicality, there is a greater probability of flowers in a particular neighbourhood getting pollinator than the more distant flowers. On account of this, the probability switch (p) is designed to have a slight bias for local pollination.

The implementation procedure is explained below:

- Step 1:** Let $P^0 = [p_1, p_2, \dots, p_N]$ be the positions of the sensor nodes in the network. Here, position of k^{th} node $p_k^0 = (x_k, y_k)$, for the 2 dimensional model.
- Step 2:** We initialize $n-m$ random positions for the initial population of the unknown nodes.
- Step 3:** The fitness of each particle in the population is evaluated using the cost function. The minimum fitness value of the initial population is found out. This is the global best.
- Step 4:** Using the probability switch, global or local pollination is selected and the positions of the particles are changed in accordance with the update equations given: $P_k^{i+1} = P_k^i + \gamma L(G^* - P_k^i)$; for global pollination, where L is a step vector drawn from a Levy distribution.
- $$P_k^{i+1} = P_k^i + \epsilon (P_k^i - P_l^i); \text{ for local pollination}$$
- Step 5:** The new positions are then checked to determine if the solution is inside the field area (simple bounds).
- Step 6:** The fitness of the new solutions are evaluated. If found better, they are updated in the population.
- Step 7:** The best solutions at the end of the maximum number of iterations rounds is the output of the algorithm.
- Step 8:** The goodness of this estimate is then evaluated by using the Node Localization Energy (NLE) equation using the estimated coordinates and the actual sensor coordinates, to determine the localization error. The topology estimate with the least error is the closest approximation to the actual sensor network topology.

4. Lifetime Improvement Methodology

The nodes in a wireless sensor network are deployed in a random manner. The locations at which the nodes land cannot be predicted beforehand. In many cases, some geographical regions may have a higher node density in comparison with other regions. In cases where the sensor nodes are entrusted with sensing geographically relevant data like humidity or temperature, the nodes in such high density areas are observed to transmit similar data. Data transmission takes up energy of the individual nodes and cut down the overall network lifetime. The sensor node locations, that can be found

using localization algorithms described earlier, could be utilized to determine node densities of the region. The nodes that are sufficiently close together are grouped together by setting a threshold value. The assumption is that nodes in a group transmit the same redundant data. For this reason, only one member of the group is elected for data transmission. As the energy of that node is used up, another member of the group is entrusted with the task of data transmission.

The communication among the group members regarding head node selection and sleep time schedule duration occurs during the setup phase of the network, and after the exhaustion of energy of the head node. This is unlike the method that occurs in Low-Energy Adaptive Clustering Hierarchy (LEACH), wherein, the cluster head selection and subsequent communication among nodes is done during every round of the algorithm. Hence, the proposed method conserves on node energies by reducing inter node communication for the algorithm duration. The data from the group members is periodically checked to verify if they contain the same data. The flow chart for the proposed algorithm is shown in Figure 1.

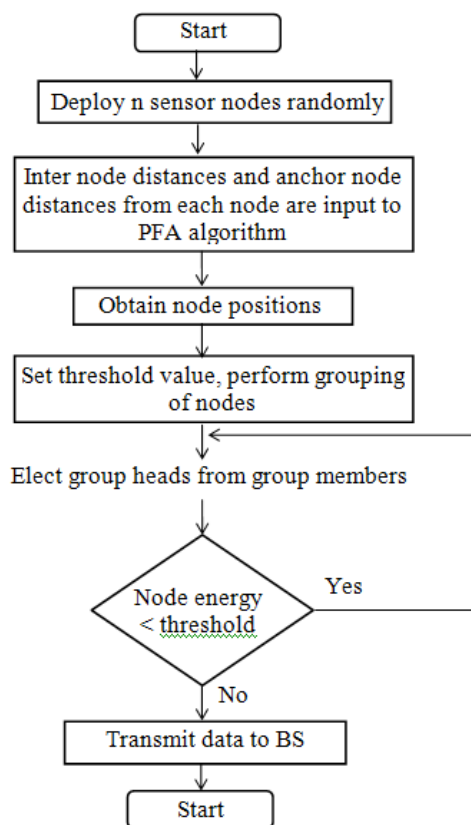


Figure 1. Flowchart for proposed method

5. Related Approaches to Localization

5.1 Particle Swarm Optimization

The authors in proposed a node localization algorithm based on Particle Swarm Optimization (PSO). PSO is an algorithm that was inspired by the flocking of birds or patterns of movement of fish. The flowchart of PSO algorithm for node localization is shown in Figure 2. The PSO algorithm searches the entire solution space and gives a set of candidate solutions referred to as 'particles'. According to this algorithm, the position and velocity of each particle is governed by a set of equations that factor on the estimate of its best location and of the global best among the population. The personal best and global

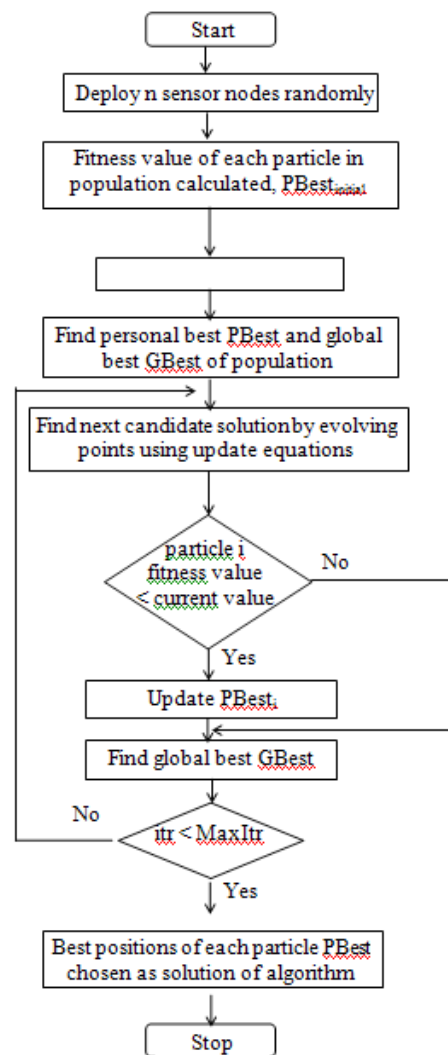


Figure 2. Flowchart of PSO process for node localization

best position of the population is found by computing the value of the cost function at the corresponding particle location. The algorithm then moves each particle towards the directions of personal best positions and global best positions with randomly weighted averages determined by Rand1 and Rand2 variables in the update equations. The PSO algorithm is a computationally efficient global optimization technique with the advantage of not getting trapped in local minima.

We assume $p_k = (p_{k1}, p_{k2}, \dots, p_{kN})$ be a vector of N dimensions which represents positions of the k th particle in the swarm, $P_{best} = [P_{k1}, P_{k2}, \dots, P_{kN}]$ the personal best of the k th particle and $G_{best} = [G_1, G_2, \dots, G_N]$ the global best of the swarm. Each particle of the swarm is evolved by the Equation. (3) and (4).

$$v_k^\theta = \omega v_{kd}^\theta + C_1 \text{rand}_1 (Pb_k^\theta - p_k^\theta) + C_2 \text{rand}_2 (Gb^\theta - p_k^\theta) \quad (3)$$

$$p_k^\theta = p_k^\theta + v_k^\theta \quad (4)$$

Where $\theta \in (x, y)$, $k=1, 2, \dots, N$ and N is the population size of the swarm. v_k and p_k are the velocity and particle position of the k th particle. C_1 and C_2 are called cognitive and social scaling parameters. C_1 determines the extent to which the particle is influenced by its personal best position, whereas, C_2 determines the extent to which the global best position of the swarm influences its movement.

The minimum value of the objective function that has to be obtained, in this case, is the square of the range errors between non-anchor nodes and the anchor nodes. This can be represented mathematically in Equation (5).

$$CF_{PSO} = \sum_{j=1}^N \frac{1}{N} \sum_{i=1}^N \left(\sqrt{(x - x_i)^2 + (y - y_i)^2} - \hat{d}_i \right)^2 \quad (5)$$

Where $\hat{d}_i$ is the corresponding noisy estimate of the distance measured from the non-anchor node to its neighbouring anchor nodes? (x_k, y_k) are the 2 dimensional coordinates of the k th anchor node and (x, y) are the target nodes locations.

5.2 Harmony Search Algorithm

Harmony Search (HS) is a music-inspired meta heuristic optimization algorithm that was developed in 2001. When observing musicians, it was found that the aim of music is to search for a perfect state of harmony, hence the inspiration for the development of this algorithm. The harmony

in music is analogous to finding the optimal value of an optimization process¹⁹. The key feature of this algorithm is the method used to find new solution in the search space. This was developed from the technique used by a musician when he is improvising a new note. To improvise a new note, he has three choices: 1. to play a famous piece of music exactly from memory; 2. adjust the pitch slightly to play something similar to a known piece or; 3. compose new random notes.

To elaborate, the HS algorithm operates on a set of K candidate solutions (or melodies). This set is called harmony memory. Extending this concept into the localization scenario, each melody represents the position of all nodes in the network, that is, it encodes the entire topology of the network. It can be denoted as, where each melody consists of n notes (positions of n nodes). The first m pairs represent the coordinates of the anchor nodes, which are known beforehand²⁰. They serve as the reference points against which the estimates may be evaluated. The value of the cost function for each melody is determined each iteration and their values are updated in a database. These K melodies are refined iteratively through the improvisation process described earlier. After the improvisation process, the value of the cost function is evaluated for each improvised melody and compared against the existing cost function values of K melodies and the best melodies are kept for further iteration rounds. This refinement is done until the maximum number of iteration rounds have been performed²¹.

The three operators used in this algorithm are:

- Harmony Memory Considering Rate (HMCR) is the probability that the new note is selected from a set of values of the same notes from K melodies in the harmony memory.
- Pitch Adjusting Rate (PAR) represents the probability that the new note is selected from a slightly altered version (pitch altered slightly) of the melody. This takes into consideration the geometrical constraints imposed by the anchor nodes on the non-anchor nodes.
- Random Selection Rate (RSR) is defined as the probability of picking an entirely random note from the search space. This is a global operator.

The objective function to be minimized in this algorithm is the sum of the squared error between the estimated position and the actual measured inter node distance, represented mathematically as

$$CF_{HS} = \sum_{i=m+1}^n \left(\sum_{j \in N} (d_{ij} - \hat{d}_{ij})^2 \right) \quad (6)$$

Where d_{ij} and $\hat{d}_{ij} = \sqrt{(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2}$ and are the measured and estimated inter node distances between nodes i and j , respectively. The flow chart for node localization using harmony search algorithm is shown in Figure 3.

6. Simulation Results and Discussion

The simulation environment was set up in MATLAB, with 100 sensor nodes deployed randomly in a 100 x 100

regular square area. 10 percent of the total number of nodes deployed was taken as anchor nodes (i.e. 10 in this case). The localization experiment was conducted for a maximum of 500 iterations and the NLE of the population at the end of each round was calculated. The parameters used for the experiments are listed in the Table 1.

The final node position estimates of the FPA localization algorithm at the end of 500 iterations is depicted in Figure 4. It is observed that the algorithm gives fairly good approximations of the sensor node locations. Most of the estimated locations are found to be in the exact locations as the actual sensor nodes. Only a few nodes are off the

Table 1. Parameters for localization algorithms

Sensor Field Size	100m x 100m		
Total number of Nodes	100		
Number of anchor nodes	10 % of total nodes = 10		
	FPA	PSO	HSA
(number of iterations) Itr = 500	Itr = 500	Itr = 500	Itr = 500
Switching probability (p) = 0.6	w = 0.7298	HMS = 6	
	$c_1 = 1.496$	HMCR = 0.95	
	$c_2 = 1.496$	PAR = 0.8	
	K = 100		

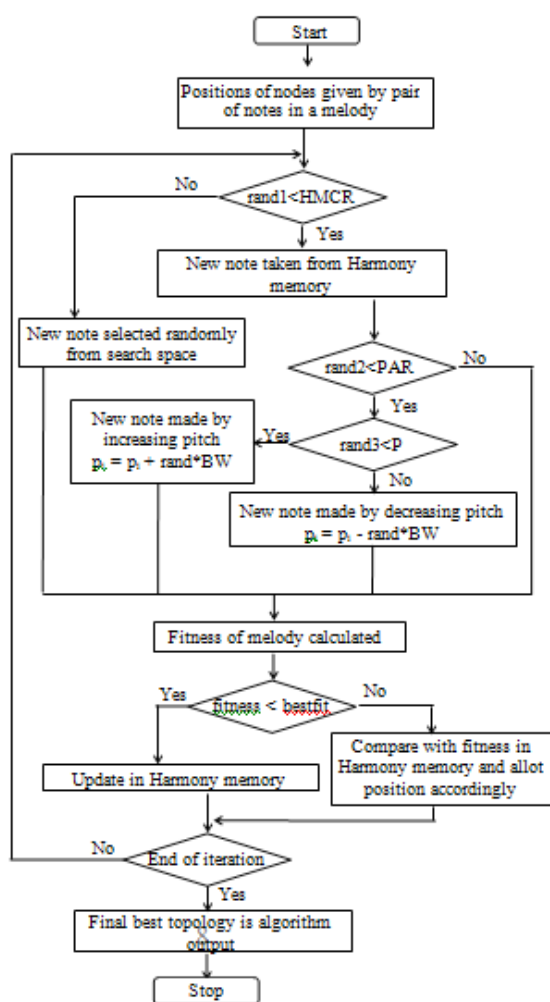


Figure 3. Flowchart of HS process for node localization

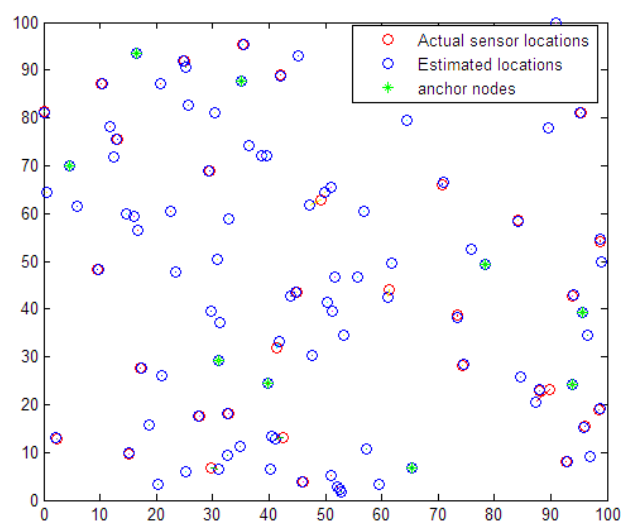


Figure 4. Localization output of the FLA algorithm at end of 500 iterations

exact mark, by only a small distance. The plot of localization error (NLE) for each iteration round for the different algorithms is given in Figure 5. The performance of the different algorithms can be compared with this graph. The NLE is observed to begin at approximately 60 and gradually converges to a minimum value for each algorithm. This convergence signifies the minimizing of the error in estimating the exact locations of the nodes. The objective here is to minimize the error to zero. From the observed convergence of the algorithm, this can be done by increasing the number of iterations to a larger number. This, however, takes much time and computational resources. The algorithm with the best convergence is then taken as the best, since better results could be obtained within a reasonable amount of time, without excessive computational burden.

The FPA method shows a quick convergence to the solution as compared to PSO and HSA. While the PSO had a much quicker convergence during the initial rounds, it eventually faded off to a gradual convergence, thereby performing better than HS by the end of 500 iterations but with a little more error as compared to FPA method. The HSA, being a population-based algorithm, performed the optimization at a slower pace, when compared to its particle-based counterparts. It is observed that from Figure 5 proposed method retains more residual energy than classic LEACH algorithm does. Its performance can be

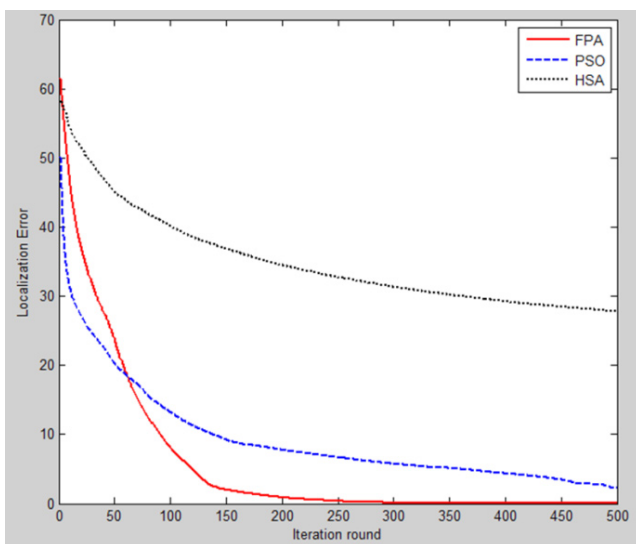


Figure 5. NLE plots of different localization algorithms for 500 iterations

calculated to be 35% better, based on the residual energy of the network.

6.1.1 Scenario 1: Transmission of 4 Kb Data

In this scenario, the nodes of both algorithms transmit 4 Kb of data through its network. It is observed from Figure 6 that the proposed method retains much more energy than LEACH algorithm does. In this scenario, the nodes of both algorithms transmit 4 Kb of data through its network. This means that each node in the sensor field transmit a data of 4096 bits in every data transmission to the BS. This size of data for each node does not include the data overhead needed for inters node communication and schedule relaying mechanism. Each node in the sensor field has an energy of 0.5 Joules, hence the algorithms start with an initial residue energy of $50 \times 0.5 = 25$ Joules. It is observed from Figure 5 that the PGA method retains much more energy than LEACH algorithm does. The slope of the graph of PGA is more gradual than that of LEACH. The final residual energy of LEACH at the end of 1500 rounds was measured to be 0.0750, while that of PGA was 17.91, as given in Table 2.

Table 2. Summarizes the simulation results of scenario 1.

	First node death (round)	Number of dead nodes	Residual energy (in Joules)
LEACH Algorithm	328	48	0.0750
PGA Algorithm	561	12	17.91

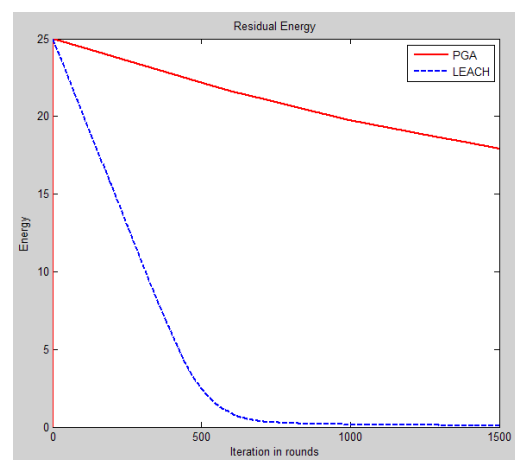


Figure 6. Residual energy plots of LEACH and PGA; scenario 1

It is also observed from Figure 7 that the first node death occurs at a later iteration round as compared to LEACH, and the total number of dead nodes at the end of 1500 iterations is considerably low as compared to that of leach algorithm. Figure 7 shows the plot of the number of dead nodes for each iteration round. It is observed that the first node death of PGA occurs at a later iteration round (561) as compared to LEACH (328), and the total number of dead nodes of PGA at the end of 1500 iterations is considerably lower (12) as compared to that of LEACH algorithm (48). Comparing the residual energy of the networks of both algorithms, the PGA is found to be 35% better than LEACH.

6.1.2 Scenario 2: Proposed Method Transfers 16Kb of Data

In this scenario, the nodes in the proposed method transmit 16 Kb of data through its network, while the LEACH algorithm performs, as before, with 4 Kb of data. The energy used by the proposed method is increased, but it is observed to be less than that consumed by the LEACH algorithm, as depicted in Figure 8.

From Figure 9, it is observed that although the first node death occurs earlier in the proposed method, the rate at which death occurs is lower than that of LEACH algorithm; hence the total number of dead nodes at the end of 800 iterations is less in the proposed method.

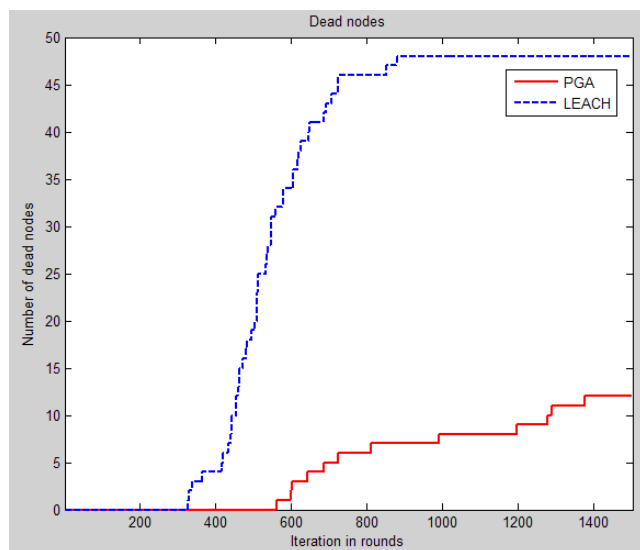


Figure 7. Plots of number of dead nodes of LEACH and PGA; scenario 1

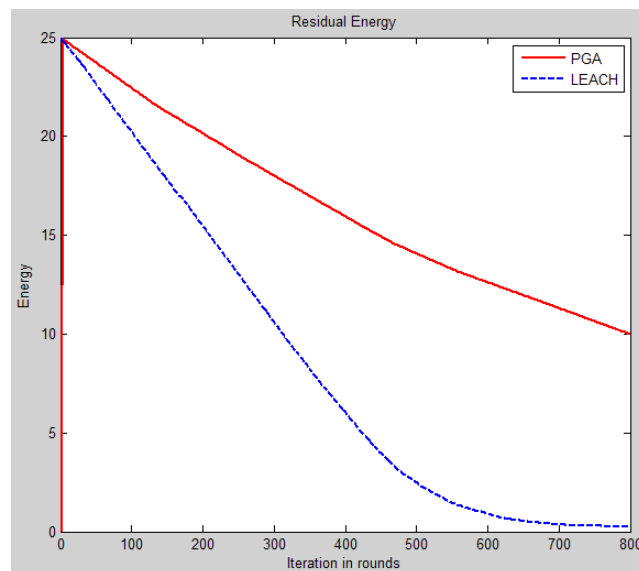


Figure 8. Residual energy plots of LEACH and PGA; scenario 2

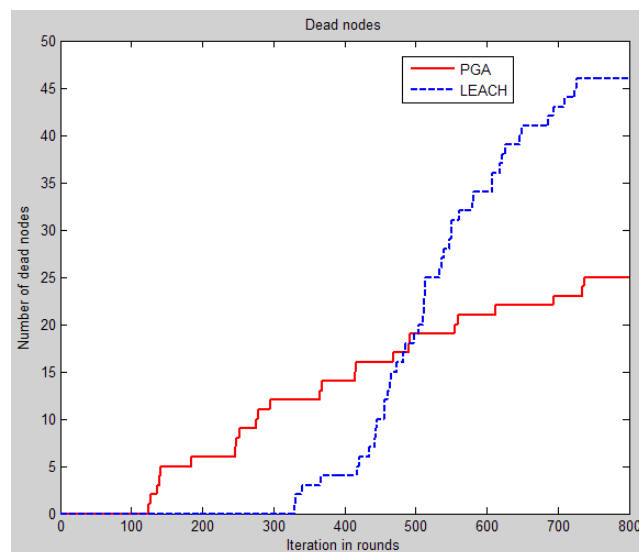


Figure 9. Plots of number of dead nodes of LEACH and PGA; scenario 2

Table 3 summarizes the results of the PGA simulation in scenario 2. According to the performance metrics, the first node death of PGA occurs at an earlier iteration round, but however, the total number of dead nodes at the end of 800 rounds is much less than that of LEACH. The residual energy is comparatively higher for PGA.

The results of the localization experiment demonstrate that FPA, for the purpose of node localization, performs

Table 3. Comparison of PGA and LEACH; scenario 2

	First node death	Number of dead nodes	Residual energy (in Joules)
LEACH Algorithm	328	48	0.0750
PGA Algorithm	124	25	9.97

better in comparison with localization done with PSO and HSA. This is observed by the fast convergence of the NLE graph. For the lifetime improvement experiment, from the simulation results, it is observed that the proposed PGA method retains more residual energy than classic LEACH algorithm does. Its performance can be calculated to be 35% better, based on the residual energy of the network.

PGA allows for transmission of higher volumes of data, and still retains more energy than LEACH does. This scenario assumes, from a more practical point of view, that the nodes are deployed in a highly dense region, and therefore, there are many nodes in a group, that allow for much conservation of energy. The data from adjacent nodes is periodically checked to verify that the data are similar. This additional data overhead is incorporated into the data transmitted by the head node. Alternatively, this method allows for the transmission of other sensor measurements from the environment with similar redundant data characteristics. Hence, the proposed PGA method is an improvement to the existing system.

7. Conclusion

The localization problem of sensor nodes in a wireless sensor network is an optimization problem. In this paper, a novel nature-inspired meta heuristic algorithm for the localization process was implemented. The computational efficiency and fast convergence of the proposed FPA method as compared to the other methods are noteworthy features. The positions of the sensor nodes determined by the algorithm are then used for the purpose of grouping nodes based on their proximity, so as to eliminate the redundant transmission of similar data. The proposed PGA method showed promising results in terms of network lifetime improvement, and can be a subject of further research.

Future work may be directed towards implementing multi-objective FPA optimization for node localization, taking into consideration the node connectivity information. This would sufficiently eliminate the additional errors due to flip ambiguity. Further flower pollen algorithm design parameters can be tuned and the algorithm could be hybrid with other intelligent nature and bio inspired algorithm for better performance in terms of accuracy and convergence.

8. References

1. Gopakumar A, Jacob L. Localization in Wireless Sensor Networks using Particle Swarm Optimization. *IET International Conference on Wireless, Mobile and Multimedia Networks*, 2008 Jan, p.227–30.
2. Kumar CC. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*. 2003 Aug; 91(8):1247–56.
3. Karthikeyan A, Sarkar S, Gupte AM, Srividhya V. Selection of Cluster Head Using Fuzzy Adaptive Clustering for Energy Optimization in Wireless Sensor Network. *Journal of Theoretical and Applied Information Technology*. 2013 Oct; 53(1):6–13.
4. Sasikumar P, Shankar T, Khara S. Balanced Cluster Head Selection Based on Modified k-Means in a Distributed Wireless Sensor Network. *International Journal of Distributed Sensor Networks*. 2016 Mar; 12(3):1–11.
5. Shankar T, Shanmugavel S, Karthikeyan A, Dhanabal R. Selection of Cluster Head Using Neural Network in Wireless Sensor Network. *European Journal of Scientific Research*. 2012 Aug; 83(3):320–73.
6. Shankar T, Shanmugavel S, Karthikeyan A. Hybrid Approach for Energy Optimization in Wireless Sensor Networks Using PSO. *IRECOS International Review on Computers and Software*. 2013; 8(6):1454–59.
7. Shankar T, Shanmugavel S, Karthikeyan A. Modified Harmony Search Algorithm for Energy Optimization in WSN. *International Journal on Communications Antenna and Propagation*. 2013 Aug; 3(4):214–20.
8. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: Harmony search. *Simulation*. 2001 Feb; 76(2):60–68.
9. Yick J, Mukherjee B, Ghosal D. Wireless sensor network survey. *Computer Networks*. 2008 Aug; 52(12):2292–2330.
10. Amundson I, Koutsoukos X. A survey on localization for mobile wireless sensor networks. *Proceeding MELT'09 Proceedings of the 2nd international conference on Mobile Entity Localization and Tracking in GPS-less*. 2009 Sep, p.235–54.
11. Kennedy J, Eberhart R. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*. 1995 Nov- Dec, 4, p.1942–48.

12. Lee KS, Geem ZW. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering*. 2005 Sep, 194(36-38), pp. 3902–3393.
13. Yang XS. Flower pollination algorithm for global optimization in Unconventional Computation and Natural Computation. *Lecture Notes in Computer Science*. 2012 Sep, p. 240–49.
14. Manjarres D, Ser JD, Lopez SG, Vecchio M. A novel heuristic approach for distance- and connectivity-based multi hop node localization in wireless sensor networks. *Springer Soft Computing*. 2013 Jan; 17(1):17–28.
15. Balasubramani K, Marcus K. A Study on Flower Pollination Algorithm and its Applications. *International Journal of Application or Innovation in Engineering and Management*. 2014 Nov; 3(11):230–35.
16. Manjarres D, Ser JD, Lopez SG, Vecchio M. On the Application of a Hybrid Harmony Search Algorithm to Node Localization in Anchor-based Wireless Sensor Networks. *IEEE Conference, on Intelligent System Design*, 2011 Nov, p. 1014–1019.
17. Kulkarni R. Bio-inspired node localization in wireless sensor networks. *Systems, Man and Cybernetics*, 2009. SMC 2009. *IEEE International Conference*, 2009 Oct, p.205–10.
18. Prathiba R, Moses MB, Sakthivel S. Flower Pollination Algorithm Applied for Different Economic Load Dispatch Problems. *International Journal of Engineering and Technology (IJET)*. 2014 Apr-May; 6(2):1009–16.
19. Shankar T, Karthikeyan A, Sivasankar P, Neha RR. Implementation of Smart sleep mechanism and hybrid data collection technique for maximizing network Lifetime in WSN's. *Indian Journal of Science and Technology*. 2015 May; 8(S9):1–8.
20. Syed Abdul Syed S, Senthil Kumaran T, Sarfaraz Ahmed A. An energy efficiency distributed routing algorithm based on HAC clustering method for WSNs. *Indian Journal of Science and Technology*. 2014 Nov; 7(S7):66–75.
21. Bhuvaneswari PTV, Vaidehi V. Enhancement techniques incorporated in LEACH- a survey. *Indian Journal of Science and Technology*. 2009 May; 2(5):36–44.