ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 8(31), DOI: 10.17485/ijst/2015/v8i31/67726, November 2015* ISSN (Online) : 0974-5645

A Case Study of Bio-Optimization Techniques for Wireless Sensor Network in Node Location Awareness

D. Chandirasekaran* and T. Jayabarathi

School of Electrical Engineering, VIT University, Vellore – 632014, Tamil Nadu, India;
chasan2k@gmail.com, tjayabarathi@vit.ac.in

Abstract

Background: In wireless sensor networks the sensors are deployed randomly in the sensing field, therefore the location awareness of the deployed nodes is challenging. The objective is to estimate the location of the deployed sensor nodes through bio optimized algorithms. **Methods:** This paper compares performance of three best bio-optimization algorithms available: Particle Swarm Optimization (PSO), Shuffled Frog Leaping (SFLA) and Firefly Algorithms (FFA) in estimating the optimal location of randomly deployed sensors. The optimum solution helps to ensure the maximum coverage of sensing capability and Quality of Service (QoS) of the network. The simulation is done using LabVIEW to understand the performance of these algorithms. **Findings:** The objective function and fitness value is calculated for the algorithms and based on those the error value is determined for 50 nodes and 10 beacons in a 100x100 sensor field dimension. To evaluate the performance under noisy environment a noise of 2% and 5% is added and simulated. Also the transmission radius of beacons changed from 25 to 20m to analyse the error value for optimum location estimation, however when changing the transmission radius it is found that the number of localized nodes are reduced. The performance is analysed based on localization error, computing time and memory. SFLA offers better in localization error however its computation time is more. The PSO takes less computing time and memory compared to FFA. The average error Vs sensor nodes depicts SFLA performance is superior. The methods adapted in this paper throw more lights on performance of these algorithms under noisy environment and effect of localization error under various transmission radiuses. **Improvements:** These algorithms can be further analysed for centralised localization and also location awareness for mobile nodes.

Keywords: FFA, PSO, QoS, RSSI, SFLA, WSN

1. Introduction

In localizing the deployed sensor nodes in wireless sensor networks bio-inspired algorithms methods of optimization[1] plays a vital role owing to their computationally efficiency compared to the conventional analytical methods. In particular Particle Swarm Optimization (PSO), Shuffled Frog Leaping Algorithm (SFLA) and Firefly Algorithms (FFA) are popular multi-dimensional optimization techniques which are easy to implement, comparatively more accurate solutions, computational efficiency and fast convergence.

In order to evaluate a WSN node localization performance we assume a network consisting of N number *of* nodes is deployed in a sensor field randomly and between the nodes "r" is the communication range. If we assume the sensors are deployed in a defined area A={X, Y} then the set of nodes deployed in the {X, Y} area will be {s1, s2….sN}. If the sensor deployed on a target point (x1, y1) can cover a location point (x2, y2), then the Euclidean distance between these two points is

$$(x1 - x2)^2 - (y1 - y2)^2 \leq r^2 \quad (1)$$

** Author for correspondence*

The existing location awareness approaches[2] are Received Signal Strength Indicator (RSSI) is the most popular method of measuring the node position by calculating the distance of nodes. Time of Arrival (ToA) and Angle-of-Arrival (AoA), Triangulation and Maximum Likelihood (ML) estimation are the other methods. RSSI technique is based on the receiving power and attenuation of radio signal exponentially with the increase of distance. In RSSI the distance can be calculated based on the loss in power by comparing the theoretical model. Time based methods Time of Arrival (ToA) and estimates the distance by the difference of propagation time between two nodes with known velocity of signal propagation. Angle-of-Arrival (AoA) also known as Direction of Arrival (DoA) techniques calculates the position by geometric coordinates with the angle from where signals are received. As per as accuracy of determination is concerned ToA, and AoA methods are ahead RSSI, due to loss in radio signal amplitude by environmental factors. Triangulation technique is based on the direction measurement of the node instead of the distance measured in AoA systems. The node positions are determined by trigonometry laws of sinø and cosinø. Maximum Likelihood (ML) estimation calculates the position of a node by minimizing the differences between the measured distances and estimated distances.

In this paper, performance study of three important bio-inspired optimization algorithms are carried out for Particle Swarm Optimization (PSO) Shuffled Frog Leaping Algorithm (SFLA) and Firefly Algorithm (FFA). All these three algorithms are analysed using LabVIEW for the suitability of node localization in WSN based on the literature revised. PSO[3], SFLA[4] and FFA[5].

The Performance comparison of shuffled frog leaping and firefly algorithm in LabVIEW[6] indicates the optimal node identification using LabVIEW simulation and similarly the pros and cons of the bio inspired algorithms for autonomous deployment of sensor nodes and localization for WSN is presented[7]. The location routing protocol that uses smart antennas to estimate nodes positions into the network and to deliver information basing routing decisions on neighbour's status connection and relative position, named LBRA[8]. In SFLA the average distance per hop is designed and location optimization is determined through PSO[9].

Robust positioning algorithm experimented in this article[10] produced the average connectivity of 12 nodes and 10% anchors and 40% errors in distance measurements. Component based localization is explained[11] by grouping nodes into components to share ranging and anchor knowledge in a better way; also it relaxes the node order and the distribution of anchor. Article[12] discussed and compared semi definite programming, simulated annealing and two-phase stochastic optimization—a hybrid scheme to perform node localization.

The advantage of distribute localization techniques over the centralised one is because of the complexity in nature and scalability issues present in centralised WSN techniques. The distributed localization algorithms will be developed on each individual sensor node instead of central base station adopted in centralised techniques. The target nodes localize based on distance measurement from the neighbouring beacons or already localised nodes. The case study done in this paper infers few features for in particular the localisation accuracy and the iterative method of localization ensures more number of nodes are localised in short span of time.

2. Bio-Inspired Optimization Techniques in WSN Localization

Natural living organism provides rich source of ideas for computer scientists. The bio-inspired algorithms offer better accuracy and modest computational time. PSO, SFLA and FFA bio inspired algorithms are discussed in the following subsections.

2.1 Particle Swarm Optimization Algorithm (PSO)

Particle swarm optimization (PSO) is a computational method that optimizes problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It is developed based on social behaviour of a flock of birds. PSO optimizes a problem by having a population of candidate solutions with particles, by using simple formulas moving these particles around in the search space over the particles positions and velocity. There have been many modifications since after its introduction[3], and many versions of PSO have been proposed and applied to solve optimization problems in diverse fields[13]. In PSO the movement of each particle is influenced by its local best known position, which will be considered as better positions found by other particles.

As mentioned earlier the two mathematical formulas are used to resolve the particle movements in the search space. The movements of the particles in the search space is guided by their own best known position and when improved positions are being discovered then these will come to guide the movements of the swarm. This process will be repeated until a satisfactory solution is obtained. The particles move around in an n-dimensional space to search the global solution, where n represents the number of parameters to be optimized, x and y coordinates of a nodes. The objective is to determine the fitness of the particle in the search space which is decided based its closeness to the global solutions. Now each particle i has a position Xi_d and moves with a velocity Vi_d, $1 \leq i \leq s$ and $1 \leq d \leq n$. The best particle which has highest fitness position in that particular iteration of the search is called $pbest_{id}$ (local best), and $gbest_d$ (global best) is the maximum of $pbest_{id}$ of all particles in the iterative search which is the best possible solution. The velocity Vid and position Xid of each particle in k^{th} iteration is updated using equation (2) and (3).

$$vi(k+1)=w.vi(k)+C1.rd1.(pbesti-xi)+C2.rd2.(gbest-xi) \quad (2)$$

$$xi(k+1)=xi(k)+vi(k+1) \quad (3)$$

Where;

vi :	Velocity of particle
xi :	current position of the particle
w :	weighting function,
c1,c2 :	weighting factor
rd1, rd2:	random numbers (0 to1) with a uniform distribution.
pbest:	Best position of the particle in that particular iteration
gbest :	Best position of the particle in the group.
i :	Iteration

2.1.1 PSO Algorithms Pseudo Code

Input: Randomly initialized particle position and velocity
Output**:** optimum position of the particle
1: Initialize w, c1, c2 and iteration maximum kmax
2: Initialize Xmin, Xmax, Vmin and Vmax
3: Initialize the target fitness fT
4: for each particle i do and for each dimension d do
5: Initialize xi randomly: Xmin ≤ xi ≤ Xmax
6: Initialize vi randomly: Vmin ≤ vi ≤ Vmax
7: end for
8: Iteration k = 0
9: while (k ≤ kmax) and (f (gbest) >fT) do
10: for each particle i do
11: Compute f (xi)
12: If f (xi) <f ($pbest_i$) then
13: for each dimension d do
14: $pbest_i$ = xi
15: end for
16: end if
17: if f(Xi) < f(gbest) then
18: for each dimension d do
19: gbestd = Xid
20: end for
21: end if
22: end for
23: for each particle i do
24: for each dimension d do
25: Compute velocity vi(k + 1) using (2)
26: Restrict vi to vmin ≤ vi ≤ vmax
27: Compute position xi(k + 1) using (3)
28: Restrict xi to xmin ≤ xi ≤ xmax
29: end for
30: end for
31: k = k + 1
32: end while

2.2 Shuffled Frog Leaping Algorithm (SFLA)

Shuffled frog leaping algorithm is swarm intelligence based biological evolution algorithm. The algorithm simulates a group of frogs in which each frog represents a set of feasible solutions. The different memeplexes are assumed as different culture of frogs which are located at different places in the solution space In article[14,15,16] the execution of the algorithm, In order to form a group "F" frogs are generated and for a N-dimensional optimization problem, frog "i" of the group is represented as Xi = (x1i; x2i; ...;xNi). Then based on the fitness values the individual frogs in the group are arranged in descending order, to determine Px the global best solution. The group is divided into m ethnic groups and each ethnic group includes n frogs by satisfying the relation F = m _ n. The ethnic group divided such that teach group will be in to their sub group like first group in to first sub group and second will be in second sub group and so on similarly frog m into sub-group m, frog m + 1 into the first sub-group again and so on, until all the frogs are divided the objective is to find the best frog in each sub-group, denoted by *Pb* and worst frog *Pw* correspondingly. The iterative formulas are as shown below

*D=rand(˦) * (Pb-Pw)*

$Pnew_w = Pw+Di; -Dmax \geq Di \geq Dmax$

Where

- r and (): Random number (0 to 1)
- Pb and Pw: The position of the best and worst frog,
- D: The distance moved by the worst frog,
- Pnew_w: the better position of the frog,
- Dmax: The step length of frog leaping.

In the SFLA algorithm execution, if the updated *Pnew_w* is in the feasible solution space m then the corresponding fitness value of *Pnew_w* will be calculated. If the resultant fitness value of *Pnew_w*is worse than the corresponding fitness value of *Pw*, then *Pw* will replace *Pb* in equation (3) andre-update *Pnew_w*. If there is still no improvement, then randomly generate a new frog to replace *Pw;* repeat the update process until satisfying stop conditions

2.2.1 SFLA Algorithms steps

1. Initialize groups and parameters such as group total number of particles N, total number of frogs N1, number of sub-groups m, number of frogs in each sub-group and the updates within the sub group
2. Analyze the initial fitness values of the particles and save the initial best positions and best fitness values, then sort all N particles in ascending order as per the fitness values;
3. According to the sub group division rule sort the N frogs in ascending order and divide them into sub-groups.
4. Find out the best fitness individual Pb and the worst fitness individual Pw of each subgroup in frog group and also the group best individual Px
5. Progress the worst solution within a specified number of iterations based on equations (3) and (4).
6. According to the fitness value, arrange particles of the group in ascending order and re-mix the particles to form a new group.
7. If stop conditions are satisfied (the number of iterations exceeds the maximum allowable number of iterations or the optimal solution is obtained), the search stops, and output the position and fitness value of the first particle of the group; otherwise, return to step (3) to continue the search.

2.3 Firefly Algorithms(FFA)

Firefly Algorithms (FFA) are developed based on the characters inspired from fireflies. The firefly species produces short and rhythmic flashes of light and the pattern of flashes is unique for each particular species. The basic motto of such flashes is to attract mating partners and search foods. The Female flies respond to male's unique pattern of flashing within the same species. As the distance increases the intensity of light decreases for any light emitting flies which strictly follows the inverse square law. When the air absorbs light then it becomes weaker and weaker as the distance increases. Luciferin is the terms used to denote the bio-luminescence from the body of the fireflies which is a light emitting compound. The above behaviour of the fireflies made the researchers to develop an algorithm which is called firefly algorithms which serves as heuristic algorithm in computational intelligence.

In optimization problems, a firefly at particular location "x" has the brightness I of a firefly can have the relationship as I(x) ∞ f(x). The light intensity I_r varies with the distance "r" such that $I_r= I_0 e^{-\gamma r}$ and also the light intensity is proportional to the attractiveness β such that $\beta = \beta_0 e^{-\gamma r^2}$ where I_0 and β_0 are the original light intensity and attractiveness constant at r=0 respectively. However, the attractiveness β is relative; it should be seen in the eyes of the beholder or judged by the other fireflies. Thus, it will vary with the distance rij between firefly i and firefly j. In addition, light intensity decreases with the distance from its source, and light is also absorbed in the media, so we should allow the attractiveness to vary with the degree of absorption. In the simplest form, the light intensity Ir varies according to the inverse square law $Ir = I_s r^2$ where I_s is the intensity at the source. For a given medium with a fixed light absorption coefficient γ, the light intensity I vary with the distance r

The implementation of the firefly behaviour described[17] is organised based on the following assumption 1. All fireflies are unisexual, which means one firefly will get attracted to all other fireflies. 2. The attraction is proportional to their brightness and distance, hence for any two given fireflies the less bright one will try to attract brighter; however. 3. If a firefly doesn't find a bright firefly than its own then it will move randomly. The following algorithms consider as brightness as objective function including the other associated constraints along with the local activities carried out by the fireflies.

Where

i: i_{th} firefly, i 2 [1; n];

n: number of fireflies;

i: Max.generation.count of the generations of fireflies (indicates iteration limit);

Ii: Magnitude of i_{th} firefly Light Intensity; depends on the objective function f (x);

$r_{i,j}$: distance between the i_{th} and j_{th} fireflies respectively.

f : (xi) = objective function of i_{th} firefly, which is dependent on its location xi that is of d-dimension

Where "d" is the dimension of x in space that is also dependent on the context of the firefly, iteration variable (t). Intensity or the brightness "I" is proportional to some objective function f(x) and the location update equation is given by (3).

$$Xi = Xi + \beta e^{([\gamma r2ij])} (Xj\text{-}Xi) + \alpha \in i \quad (6)$$

where α is the step controlling parameter, r is the variable that brings about randomness, γ is the attraction coefficient, β is the step size towards the better solution, is a vector of random number from Gaussian distribution and Xi, Xj are the firefly are the location information of the observing entity.

2.3.1 Firefly Algorithm Pseudo Code

Begin

1: Generate initial population of firefly's with location xi,

i = 1; 2; 3: n;

2: Define objective function f (x), where x = (x1; x2; xd) T;

3: Generate initial population of fireflies xi , i = 1;2;3:::n;

4: Light intensity Ii of a firefly ui at location xi is determined by f (xi);

5: Define light absorption coefficient γ;

6: While(t < max generation) do /*for all n- fireflies*/

7: for i=1:n do /*for all n- fireflies*/

8: for j=1:i do

9: if (Ij> Ii) then move firefly i towards j in d-dimension

10: else

11: end for

12: end for

13: Attractiveness varies with the distance r via exp (-γr);

14: Evaluate new solutions and update light intensity;

15: end for

16: end while

17: Rank the fireflies and find the current best;

18: end

3. Problem Statement and Methodology

In WSN node localization the objective is to perform estimation of coordinates of the distributed nodes to know their initial locations. If there is a maximum of N target nodes then using M stationary beacons whose know their locations then the location of unknown nodes will be determined. The following study approach is formulated for the localization of the same;

1. Initialize the sensors randomly
2. Initialize the beacons randomly
3. Calculate real distance ie the actual distance between the beacon and each deployed sensor nodes
4. Assign measured distance ie the distance obtained by the beacons using ranging techniques. This is done by adding noise to the real distance.
5. Find out how many sensors are within the transmission range of 3 or more beacons
6. For each sensor that can be localized for PSO, SFLA and FFA are applied to minimize the objective function which represents the error function given by the equation (7)

$$\sum_{i=1}^{n} ei = \sum_{i=1}^{n} \frac{\left(Ri - \sqrt{(xi-xm)^2 + (yi-ym)^2}\right)^2}{Ri} \quad (7)$$

Here

Ri: the inexact ranging distance.

(xi, yi): beacon positions

(xm, ym) : position occupied by the particle

"n" : number of beacons having transmission coverage over that sensor

7. The algorithms return the closest values of the coordinates (xm, ym) such that error is minimized.
8. The algorithm is then applied to the next sensor in range
9. The sensor for which the location is found will act as a beacon and also goes out from the non-localized list.
10. Steps will be continue till all the nodes get localized and/or no nodes to localized yet. The values of "Ri" and "ei" will decide the performance, lower the better.

The number of iteration steps will increase if the main focus is to localize more number of nodes and thus increases the number of likelihoods for already localized nodes. In case of localizing the nodes the following points are to be highlighted: 1. If three references used in localizing in k iteration and in (k+1) iteration the usage of reference may increase, which ultimately decreases the issue of ambiguity. 2. Also it increases the time needed to localize a node. The issue of limiting the references are taken care off in this comparative study by limiting it to six, which is chosen arbitrarily. The simulation is done using LabVIEW graphical user interface, the advantages of using LabVIEW can help for real time implementation in future scope of research.

Simulation is done in LabVIEW to understand the performance of WSN Localization. We chose 50 nodes as target to be localized and 10 beacons. The sensor field dimension is considered as 100×100 square units and the transmission radius of beacon r = 25 units. The same simulation settings in LabVIEW for both the performance studies are made and the results are presented.

For the performance study of the optimization algorithm for PSO, SFLA and FFA the following parameters has been considered:

- Population : 70
- Number of Iterations : 60
- Constants c1=c2 : 2
- Inertial weight w : 0.1 to 0.9
- Particle position limits: Xmin=0 and Xmax=100
- Total 30 trial experiments : 30
- Percentage of noise added : 2% and 5% of

As per equation (7) the average of total localization error "ei" is computed and the error is calculated.

4. Results

The optimization algorithms analysed for their performance produces fair results. The initial deployment is kept random and same and hence it doesn't produce the same solution and hence the number nodes localized also varies in each iteration. The computing time gets affected because of the above reason; however the results of multiple trial runs (iterations) are averaged and compared. Figure 1 displays the LabVIEW simulated results of a particular trial run between beacons, actual and estimated nodes for PSO, SFLA and FFA respectively. Similarly the Table-1 is the summary of the various performance parameters obtained from different trial runs. From the result it is evident the localization accuracy is altered by adding the noise (Pn) in distance measurement and it is also observed that by changing the Pn value from 5 to 2 reduces the localization error. The performance is analysed based on three parameters: Localization error, computing time and memory requirement. SFLA performance depicts superior in terms of localization error than FFA and PSO, however the computing time required for SFLA and FFA is considerably more. In terms of memory required SFLA and FFA is better than that for PSO. Figure 2 shows up the relation between average error Vs localised sensor nodes and the beacon ratio.

The observation is also made 1. By changing the range distance of the node location within a square area; the first five trial runs out of the 50 are summarized in Table 1. 2. By varying the transmission radius the impact on each algorithm is tabulated in Table 2, It is evident that the number of non-localized nodes increases when the transmission radius is made as 20 units from 25 units. It is also found that there is a correction of error due to flip of ambiguity from the Table 1.

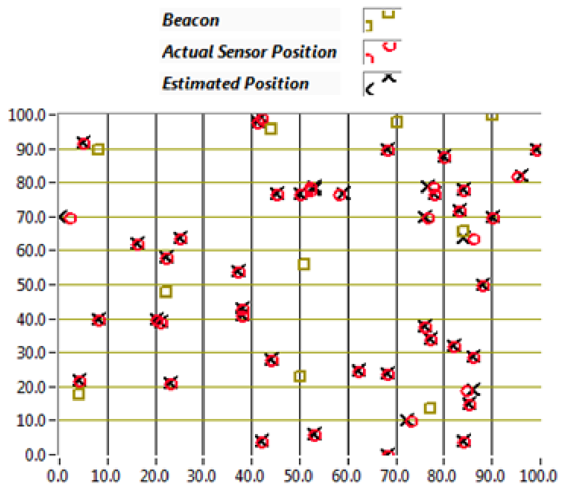
Table 1. Result of impact on ranging distance error of SFLA, FFA and PSO

Major Parameters	SFLA		FFA		PSO	
	Pn=2%	Pn=5%	Pn=2%	Pn=5%	Pn=2%	Pn=5%
Avg. no of non-localized nodes(U_l)	0.37	1.22	0.54	1.69	0.61	1.96
Avg. time taken*(s)	418.1*	383.5*	310.9*	521.5*	110.9*	121.5*
Avg. localization error (ei)	0.327	0.64	0.572	0.841	0.527	0.764

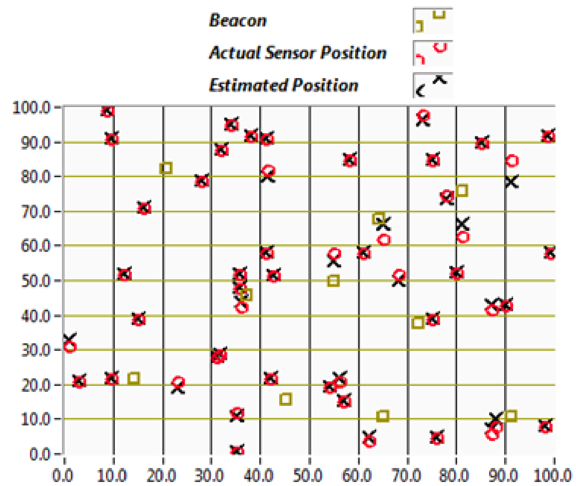
Table 2. Result of impact on varying the transmission radius (r)

Major Parameters	SFLA		FFA		PSO	
Transmission radius (r)	20	25	20	25	20	25
Avg no of non-localized nodes	1.9	0.36	2.8	0.57	3.2	0.64
Avg. time taken*(s)	281.8*	583.5*	104.4*	218.5*	40.4*	111.5*

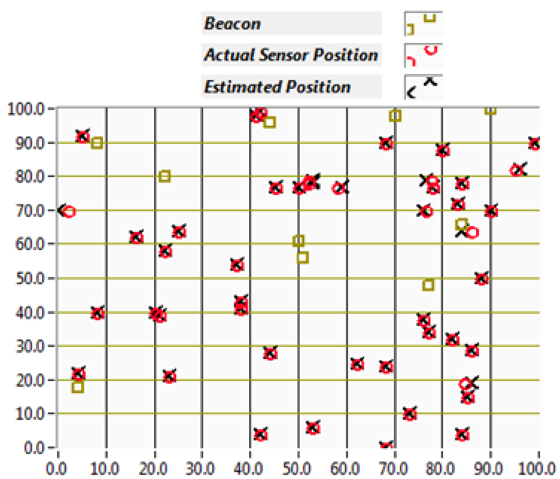
*All simulation are performed in the same computer



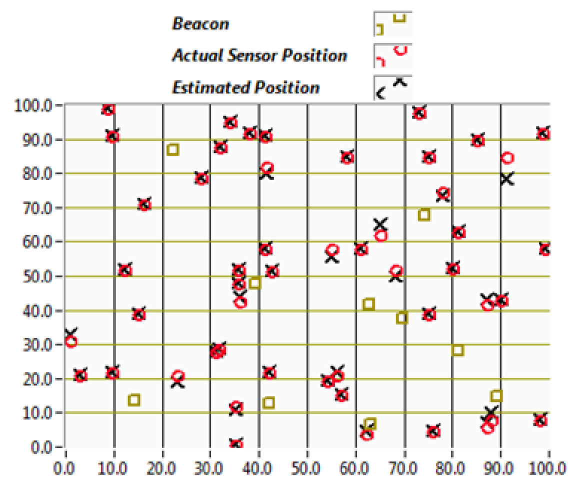
(a) Locations estimated by PSO r=25m.



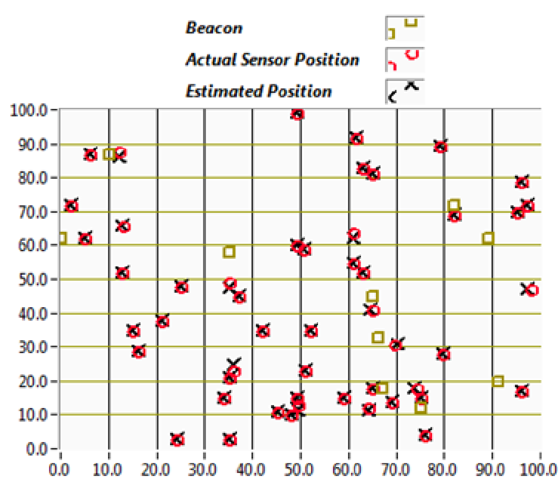
(a1) Locations estimated by PSO r=20m.



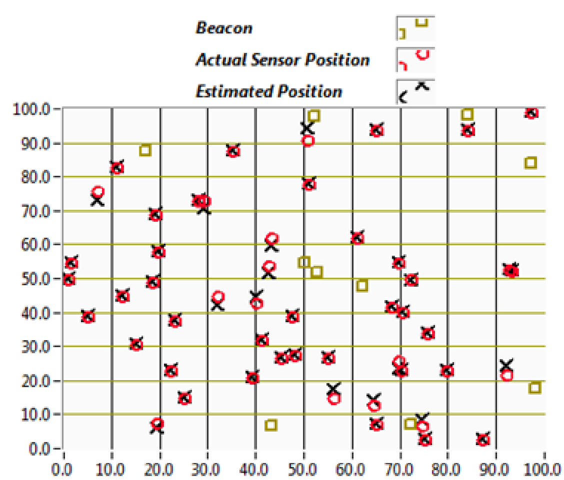
(b) Locations estimated by SFLA r=25m.



(b1) Locations estimated by SFLA r=20m.



(c) Locations estimated by FFA r=25m.



(c1) Locations estimated by FFA r=20m.

Figure 1. Result of trial run of PSO, SFLA and FFA algorithms for the same deployment with N=50; M=10; and the sensor field range is 100x100 square units.

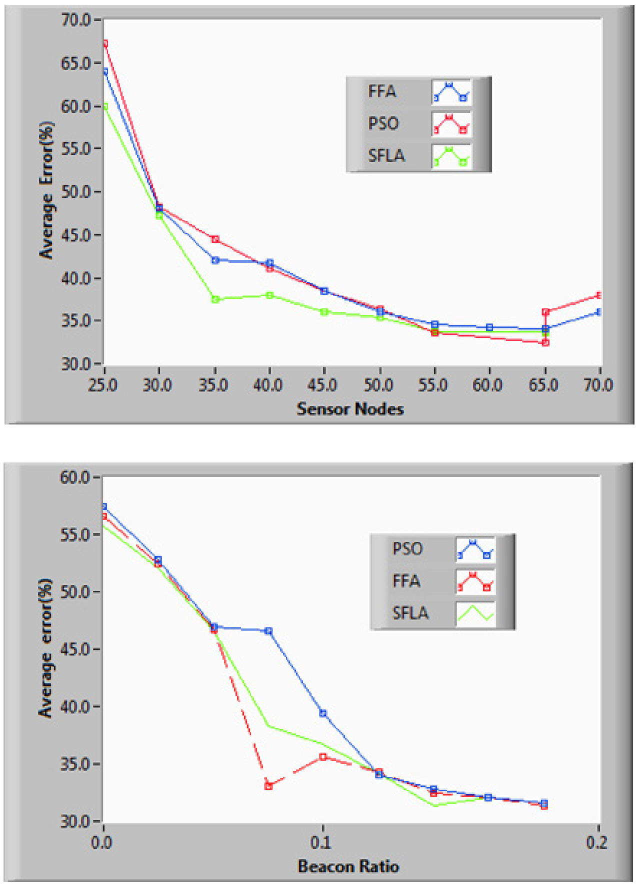


Figure 2. Result of trial run of PSO, SFLA and FFA algorithms for average error Vs Sensor nodes Vs beacon ratio.

5. Conclusion

This paper has discussed PSO, SFLA and FFA, bio-inspired algorithms to find out the localised nodes of a WSN in a scattered and iterative method. The localization problem is considered as a multidimensional optimization problem and solved by the above mentioned population-based optimization algorithms. From the results obtained it was found that SFLA and FFA offers less error value in comparison to the PSO but takes longer computational time to perform. We also ran the program with a smaller transmission radius and found that it leads to less number of nodes being localised. Although there is not vast difference in the errors offered by both the selection of what algorithms to use for localisation depends entirely on the hardware available to the user and the time constraints involved. This paper has also briefly presented a statistical summary of the results for comparison of PSO, SFLA and FFA. Both the algorithms are effective in their own way and can be further modified to suit the users need by changes in the program code to give even better results than what was obtained.

This work can be extended in a possible further study: PSO, SFLA and FFA can be used in centralized localization method so that to compare the localisation methods of centralized and distributed techniques, which can lead to solve energy awareness issue in WSN.

6. References

1. Kulkarni RV, Ganesh KV. Bio-Inspired node localisation in wireless sensor networks. IEEE International conference on system; Sant Antonio, TX. USA: 2009. p. 205–10.
2. Singh PK, Tripathi B, Singh NP. Node localization wireless sensor networks. (IJCSIT) International Journal of Computer Science and Information Technologies. 2011; 2(6):2568–72.
3. Gopakumar A, Jacob L. Localization in wireless sensor networks using particle swarm optimization. Proceedings IET International conference on Wireless, Mobile and Multimedia Networks; Mumbai, India: 2008. p. 227–30.
4. Fan X, Du F. Shuffled frog leaping algorithm based unequal clustering strategy for wireless sensor networks. Applied Mathematics and Information Sciences. 2015; 9(3):1415–26.
5. Cao S, Wang J, Yang XS. A Wireless sensor network location algorithm based on firefly algorithm. Berlin, Germany: Springer Communications in Computer and Information Science; 2012. p 18–26.
6. Chandirasekaran D, Jayabarathi T. Wireless sensor networks node localization-a performance comparison of shuffled frog leaping and firefly algorithm in LabVIEW. Telkomnika Indonesian Journal of Electrical Engineering. 2015 Jun; 14(3):516–24. ISSN: 2302-4046
7. Kulkarni RV, Venayagamoorthy GK. Bio-inspired algorithms for autonomous deployment and localization of sensor nodes. IEEE Transactions on Systems Man and Cybernetics-Part C Application and Reviews. 2010 Nov; 40(6):663–75.
8. Cobo L, Castro H, Quintero A. A location routing protocol based on smart antennas for wireless sensor networks. Indian Journal of Science and Technology. 2015 Jun; 8(11):70655.
9. Ren W, Shao C. A localization algorithm based on SFLA and PSO for wireless sensor networks. Information technology journal. 2013:12(3):502–5. ISSN 1812-5638
10. Savarese C. Robust positioning algorithms for distributed Ad-Hoc wireless sensor networks. Research project at Berkeley Wireless Research Center. 2002. p. 317–27.
11. Wang X, Luo J, Liu Y, Shanshan Li, Dezun Dong. Component-based localization in sparse wireless networks. IEEE/ACM transactions on networking. 2011 Apr;19(2):540–8.
12. Szynkiewicz E, Marks M. Optimization schemes for wire-

less sensor network localization. International Journal of Applied Mathematics and Computer Science. 2009; 19(2): 291–302.
13. Hu X, Shi Y, Eberhart R. Recent advances in particle swarm. Proceedings of CEC 2004 Congress on Evolutionary Computation. IEEE Press. 2004 Jun 19–23; 1:90–7.
14. Farahani M, Movahhed SB, Ghaderi SF. A hybrid meta-heuristic optimization algorithm based on SFLA. 2nd International Conference on Engineering Optimization; Lisbon, Portugal. 2010 Sep 6-9. p. 1–8.
15. Ebrahimi J, Hosseinian S, Gharehpetian G B. Unit commitment problem solution using shuffled frog leaping algorithm. IEEE Transactions On Power Systems. 2011 May; 26(20):573–81.
16. Sun H, Zhao J. Application of particle sharing based particle swarm frog leaping hybrid optimization algorithm in wireless sensor. Journal of Information and Computational Science. 2011; 8(14):3181–8.
17. Yang XS. Firefly Algorithm, L´evy Flights and Global Optimization. arXiv: 1003.1464v1 [math.OC]. 2009 Oct 7. p.209–18.