Wireless Pers Commun (2016) 87:1057–1069
DOI 10.1007/s11277-015-2635-4

A Comparative Analysis of Intelligent Algorithms for Localization in Wireless Sensor Networks

R. Harikrishnan[1] · V. Jawahar Senthil Kumar[2] ·
P. Sridevi Ponmalar[3]

Published online: 5 May 2015
© Springer Science+Business Media New York 2015

Abstract In a smart and decision making environment the location information of the sensors and devices under monitoring and control, is very much important, otherwise the sensed data becomes meaningless. This paper proposes three intelligent algorithms namely differential evolution localization algorithm, firefly localization algorithm, and a hybrid firefly differential evolution localization algorithm for wireless sensor networks localization problem. The proposed algorithms are range based and distributed localization algorithms. The algorithms are studied, analyzed and compared with respect to time complexity, convergence and accuracy of the estimated location information.

Keywords Differential evolution · Firefly · Wireless sensor networks · Sensor node localization · Landmarks · Sensor node

1 Introduction

Intelligent monitoring and control of home, cities, industry and environment requires an event and location based measurement and management. The measurement is done by smart and intelligent sensors. The development of technologies in processors, memory,

✉ R. Harikrishnan
rhareish@gmail.com

V. Jawahar Senthil Kumar
veerajawahar@gmail.com

P. Sridevi Ponmalar
sridevi_ponmalar@yahoo.co.in

[1] Faculty of Electrical and Electronics Engineering, Sathyabama University, Chennai, India

[2] Department of Electronics and Communication Engineering, College of Engineering Guindy, Chennai, India

[3] Faculty of Information and Communication, College of Engineering Guindy, Chennai, India

wireless communication over short distance and multi hop, very large scale integrated circuits had led to a cost effective, low powered, small sized, low data rate, self organizing smart sensors [1]. These smart sensors known as sensor nodes can be grouped together in a particular topological infrastructure arrangement to form a wireless sensor networks. The sensor node can interact with the physical environment and then sense and or control the physical parameters. The sensor nodes in a wireless sensor network can collaborate with each other to achieve a particular task in an application by using wireless communication [2].

Sensor nodes in wireless sensor networks have an architecture that can sense, compute, store, communicate and control the physical parameters of the environment and are self powered by on board batteries [3].

The wireless sensor networks follows 802.15.4 IEEE standard and can be used for various applications like precision agriculture, health monitoring, animal habitat monitoring, road traffic monitoring, forest fire detection, water quality monitoring, intrusion detection, inventory management, miners monitoring, and surveillance [4]. Most of these applications demand the location information of the sensors or devices under monitoring. The location information has to be estimated either by the sensor or by a means of central processing in the network. The sensed data would be meaningless or not in complete sense without the location information. Some data requires inherent location information like in a geographical routing application [5]. The location information leads to intelligent decision making for monitoring and controlling applications.

Some sensor nodes in wireless sensor networks may know its location and are called anchors or landmarks. This is possible by using geographical positioning system (GPS) or placing sensor nodes at known coordinate position. Other sensor nodes in the network may not know its location and they are called as location unknown sensor nodes. GPS may not be feasible for energy constraint and small sized wireless sensor network. Moreover all the areas may not be covered by GPS and it may not be cost effective for a very large deployment of sensors and further expansion of the network as the application demands [6]. The estimation of Location information of unknown sensor nodes can be estimated with the help of landmarks by inter sensor measurements such as distance, time difference of arrival, angle of arrival, radio signal strength and connectivity [7]. These measurements are used by localization algorithms to estimate the location information of unknown sensor node.

The sensor node localization algorithm can be anchor based or anchor free, range based or range free and depending on the algorithm processing the sensor node localization can be centralized or distributed. The anchor based localization algorithm requires landmarks for the location estimation where as the anchor free localization algorithms do not require anchors for location estimation. The range based localization algorithm needs distance measurement between anchor nodes or beacons and location unknown sensor nodes. The distance measurement is done by using radio signal strength, time of arrival or time difference of arrival. Whereas the range free needs only the existence of beacon signals in the medium [8]. In the centralized localization algorithm the location information is processed by a centralized processor in the sink node or base station, where as in a distributed localization algorithm the individual sensor node can compute its own location information with help of neighbouring sensor nodes and landmarks. Centralized localization algorithms have better accuracy and easy to design compared to distributed localization algorithms. Centralized localization algorithms suffer scalability problem for large sized sensor network. The distributed localization algorithm is computationally efficient and the communication of location information over multi hop to the centralized processor is not required. The energy requirement for communication is reduced and so the lifetime, reliability and overall performance of the wireless sensor networks are improved [9].

In this paper range based distributed localization algorithm is considered. The range based method uses multi lateration technique that uses three landmarks and its radio signal strength to calculate the distance between the landmark and location unknown sensor node. The aim is to design an algorithm that has, less computation time, less complex, needs less memory, easy to implement, flexible, uses less control parameters, has better accuracy and is suitable for resource constraint wireless sensor networks, without compromising the performance. This led to the motivation of developing computation intelligence algorithm for sensor node localization. Computational intelligence (CI) combines the techniques of learning, adaptability, and nature selection of evolution. Evolutionary computing makes the algorithm intelligent and provides the capability of decision making [10]. The localization of wireless sensor networks is considered as a minimization optimization problem, where the localization error is optimized for better estimation of the location information and increase the accuracy. An analytical optimization method requires more computation and the complexity increases exponentially as the network infrastructure expands. This motivated to consider bio inspired and nature inspired algorithms for resource constraint wireless sensor networks [11]. The nature adapt itself to resolve complex relationship and find the optimal solution for balancing ecosystem and maintain the diversity. The principle strategy used is simple which yields better results. Bio inspired optimization uses heuristic approach, that imitate nature and many processes in the nature can be considered as an optimization problem. Biologically inspired algorithm is based on certain phenomenon of the nature that can optimize itself for their needs and survival. Biological inspired algorithms are considered powerful and are analyzed for solving complex numerical optimization problems [12].

This paper analyses and compares three nature and bio inspired intelligent algorithms namely differential evolution localization algorithm, firefly localization algorithm, and hybrid firefly differential evolution localization algorithm in terms of time complexity, convergence and accuracy of the estimated location information.

2 Sensor Node Localization Problem Formulation

Localization scenarios of total 50 sensor nodes in a space size of 100 are considered. In this 10 sensor nodes with known location information are considered as anchor nodes or landmarks. Remaining 40 sensor nodes are considered as location unknown sensor nodes. The aim is to estimate the localization or location information of these 40 location unknown sensor nodes with the location knowledge of landmarks. Localization is estimated in two steps, ranging and position estimation. In ranging the distance between landmarks and location unknown sensor nodes are calculated using trilateration method. The position is estimated by the three nature based intelligent algorithms individually and the results of these are compared for their accuracy, convergence and computation time. Three location known landmarks which are nearer to the selected location unknown sensor node are chosen. With the help of these landmarks and its location information, the distance between the landmarks and chosen location unknown sensor node is calculated. And from this calculated distance and landmarks location information the location of unknown sensor node is estimated using the different algorithms.

First the three landmarks positions are taken. The distance between the three landmarks and location unknown sensor nodes are calculated by using the Eq. 1

$$\mathrm{d_{ist}} = \sqrt{\left((\mathrm{x_{est}} - \mathrm{x_{anc}})^2 + (\mathrm{y_{est}} - \mathrm{y_{anc}})^2\right)} \tag{1}$$

The localization function is calculated by using the Eq. 2

$$\mathrm{f(X_u, Y_u)} = \left[\sqrt{(\mathrm{X_{est}} - \mathrm{X_{anc}})^2 + (\mathrm{Y_{est}} - \mathrm{Y_{anc}})^2} - \mathrm{d_{ist}}\right]^2 \tag{2}$$

$\mathrm{X_{est}}, \mathrm{Y_{est}}$ is the estimated X and Y position of location unknown sensor node respectively.$\mathrm{X_{anc}}, \mathrm{Y_{anc}}$ is the X and Y position of anchor or landmark respectively.$\mathrm{d_{ist}}$ is the distance between anchor and location unknown sensor node.

3 Differential Evolution Localization Algorithm

Differential evolution is a stochastic search optimization algorithm that can be easily applied for experiment minimization. Differential evolution is parallel operable, flexible in implementation with less control parameters, and also has consistent convergence properties [13].

The basic operations involved in differential evolution localization algorithm are converting sensor node localization problem into differential evolution optimization problem, mutation, recombination or cross over and selection [14]. The following procedure is used in the differential evolution localization algorithm.

Localization estimation variables are selected and vector population of 20 is taken. Initialize these vector populations within a solution space. Minimization of localization error is taken as fitness function for differential evolution localization algorithm represented by Eq. 2.

Differential evolution localization algorithm evaluates fitness value for each vector in the population. Next is the Mutation process.

In differential evolution, mutation is given more importance than cross over process. The purpose of mutation is to create diversity for the solution and also mutation provides required parameter variation to get better results in time. Depending on the fitness value, target vector is selected and then mutated as per the following mutated vector Eq. 3

$$M_t^{mutated} = M_{best} + S * (T_V - R_V) \tag{3}$$

S—scaling factor = 0.7; $\mathrm{M_{best}}$—best optimal solution in the population; $\mathrm{M_t}$—mutated vector; $\mathrm{T_v}$—target vector position; $\mathrm{R_v}$—randomly selected vector position.

The next step is the cross over process that generates trail vector from target vector and mutated vector. Based on cross over constant CC = 0.6, recombination process recombines either mutated or target vector variables as per the following trail vector Eq. 4.

$$C^{trail} = \begin{Bmatrix} M^{mutated} & if(rand) \leq CC \\ T^{t\arg et} & if(rand) > CC \end{Bmatrix} \tag{4}$$

The value of scaling factor (S) and cross over constant (CC) is selected based on the Ref. [13]. And further the values are tuned according to the accuracy performance of the algorithm.

Next is selection process which does one to one selection. This selection process decides whether target or trail vector is taken for the next iteration. Initially objective function is calculated for the number of population vectors. The vector which has minimum value is the first target vector. In the selection process objective function of target vector and trail vector is calculated. This provides the fitness value of target vector and trail vector. Target vector is replaced by trail vector, provided trail vector has minimum value compared to the target vector. The selection process equation is given by the Eq. 5.

$$S_P^{n+1} = \left\{ \begin{array}{ll} T^{trail} & iff(trail) < f(t\arg et) \\ T^{t\arg et} & iff(t\arg et) \le f(trail). \end{array} \right\} \tag{5}$$

Selection process is repeated for every vector in the population to maintain population size. A maximum of 100 iterations are considered as stopping criterion for differential evolution localization algorithm. The above procedure is summarized as the following differential evolution localization algorithm steps.

Step 1: Localization estimation variables are selected as members of a vector population.
Step 2: Initialize vector population within solution space.
Step 3: Minimization of localization error is taken as fitness function of differential evolution localization algorithm.
Step 4: Using Mutation process target vector is selected and mutated to get mutated vector.
Step 5: Crossover is done on mutated vector to get trail vector.
Step 6: Selection process decides target vector or trail vector for the next iteration.
Step 7: Next iteration population is generated using selection process.
Step 8: Continue to execute step 4, step 5, step 6 and step 7 until maximum iteration is satisfied.
Step 9: Estimate the optimal location information of the sensor node.

Differential evolution localization algorithm output is shown in the Fig. 1.

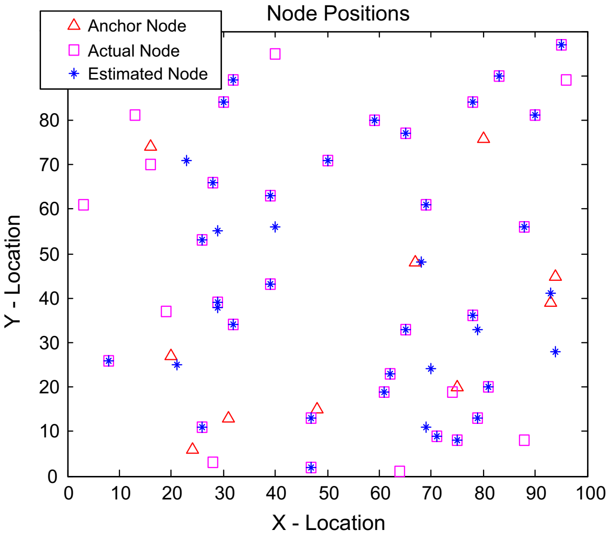

Fig. 1 Output of the differential evolution localization algorithm

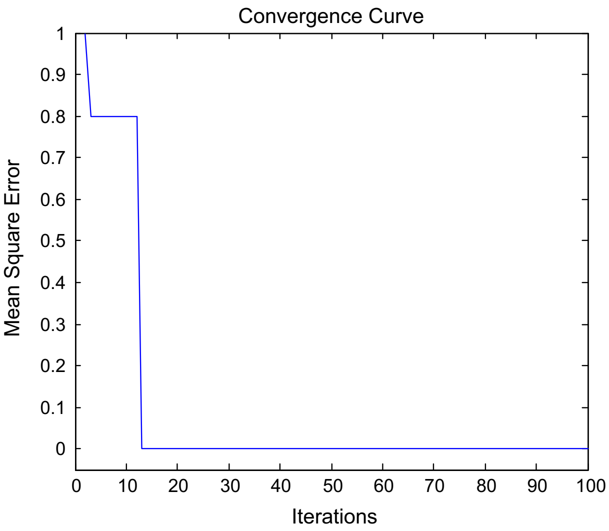

Fig. 2 Convergence curve of the differential evolution localization algorithm

Table 1 Design parameters of differential evolution localization algorithm

Parameters	Values
Maximum iteration	100
Space size	99
Landmarks	10
Unknown nodes	40
Total nodes	50
Number of vector population	20
Scaling factor, S	0.7
Cross over constant, CC	0.6

Differential evolution localization algorithm convergence curve is shown in the Fig. 2. Table 1 provides the details about the parameters considered in the design of differential evolution localization algorithm.

4 Firefly Localization Algorithm

This algorithm is based on firefly algorithm developed by Dr. Xin-She Yang in 2007 which uses flash light nature of fireflies [15].

Fireflies flash lights, to search for food by means of attracting their prey, to find their mates, and safeguard themselves from their predators for survival. Firefly algorithm depends on the intensity of light emitted by fireflies. Flashing light of firefly can be easily formulated as objective or fitness function of the sensor node localization problem. The firefly algorithm is simple in concept and easy to implement [16].

Firefly localization algorithm is based on the following ideal social behavior rules of fireflies.

1. All fireflies are unisex. One firefly move towards another brighter firefly regardless of sex.
2. Attractiveness is proportional to brightness and inversely proportional to distance. If no brighter firefly is found, firefly movement is random in nature.
3. The brightness or light intensity of the firefly is found from the sensor node localization objective function given by the Eq. 2.

Based on the above rules the firefly localization algorithm is designed [17]. The basic operations are, converting firefly algorithm into sensor node localization optimization problem, attractiveness calculation, distance calculation and position movement of the firefly. The design process of firefly localization algorithm is as follows.

Localization estimation variables are selected as fireflies and population size of 20 is taken. Initialize this vector population within the solution space. Minimization of localization error is taken as fitness function for firefly localization algorithm represented by Eq. 2. This objective function is used to find the brightness each firefly. Firefly localization algorithm evaluates fitness value for each vector in the population.

Next is the calculation of attractiveness process. Attractiveness is the ith firefly move towards more attractive brighter jth firefly. This attractiveness function is given by the Eq. 6

$$A(d) = A_0 * \exp(-d^m) \quad \text{With m} \gg 1 \tag{6}$$

d—Distance between two fireflies; A_0—Initial attractiveness at d = 0; γ—Absorption coefficient that controls the light intensity.

The attractiveness decreases with the increase in distance because of the absorption factor in the nature. Next is the distance calculation process between the ith firefly and the jth firefly. The distance of attraction of brightest firefly from the current firefly is calculated by the Eq. 7

$$\mathrm{d_{ij}} = \sqrt{\left(\left(\mathrm{a_i} - \mathrm{a_j}\right)^2 + \left(\mathrm{b_i} - \mathrm{b_j}\right)^2\right)} \tag{7}$$

$\mathrm{d_{ij}}$ is the distance between ith firefly and jth firefly in a two dimensional solution space.

Next is the movement process. The movement of ith firefly towards brighter jth firefly is based on attractiveness and distance between them and is given by the Eq. 8

$$m_i^{k+1} = m_i^k + A_o * \exp\left(-d^2\right) * \left(a_j^k - a_i^k\right) + \propto \quad * rand_i^k \tag{8}$$

m_i^k—Initial position of ith firefly; $A_o * \exp(-d^2) * \left(a_j^k - a_i^k\right)$—Attractiveness towards jth firefly by ith firefly; $\propto \quad * rand_i^k$—random movement of ith firefly.

If the firefly does not find any brighter firefly, it will move randomly. The above processes repeat up to the stopping criterion is achieved, that is when 100 iterations are over or convergence is achieved. The brightest firefly position is the estimated location of the sensor node.

The above procedure is summarized as the following firefly localization algorithm steps.

Step 1: Firefly is a set of control variables of sensor node localization function
Step 2: Initialize fireflies' population size of 20 in the solution space
Step 3: Objective function of sensor node localization is calculated to find the brightness of the fireflies using Eq. 2
Step 4: Attractiveness of firefly towards other fireflies is calculated using Eq. 6

Step 5: Distance between the fireflies calculated using Eq. 7
Step 6: Firefly i move towards brighter j firefly using Eq. 8
Step 7: Rank the fireflies and find the current global best
Step 8: Continue to execute the steps 4, 5, 6 and 7 till the stopping criterion is satisfied
Step 9: Estimate the optimal location information of the sensor node.

The output of the firefly localization algorithm is shown in the Fig. 3.

The convergence curve of the firefly localization algorithm is shown in the Fig. 4.

Table 2 provides the details about the parameters considered in the design of firefly localization algorithm.

Fig. 3 Output of the firefly localization algorithm

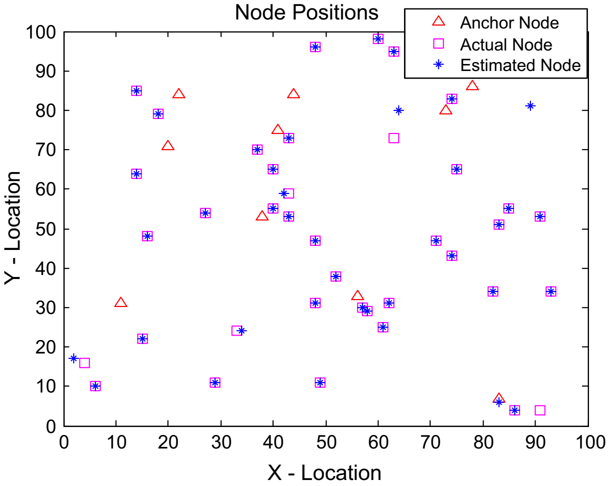

Fig. 4 Convergence curve of the firefly localization algorithm

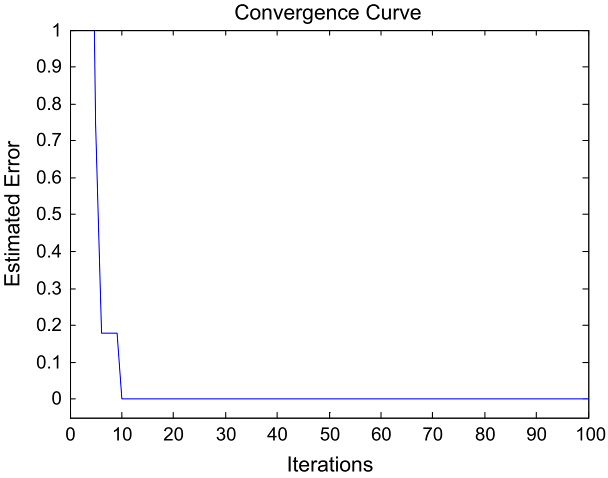

Table 2 Design parameters of firefly localization algorithm

Parameters	Values
Maximum iteration	100
Space size	99
Landmarks	10
Unknown node	40
Total node	50
Number of fireflies	20
Randomizing coefficient, α	0.2
Initial attractiveness at $d = 0$, A_o	1.0
Absorption coefficient, γ	0.96

5 Hybrid Firefly Differential Evolution Localization Algorithm

Firefly localization algorithm does not have mutation process. The performance of the firefly localization algorithm (FLA) can be improved by adding mutation process used in differential evolution localization algorithm (DELA). Differential evolution localization algorithm has better mutation process, so firefly is hybrid with differential evolution. Hybrid firefly differential evolution localization algorithm (HFDLA) is derived by using the firefly flashing light intensity, attractiveness and movement taken from FLA, and the mutation process taken from DELA. The differential evolution mutates the firefly with the randomly selected fireflies in the same population. The mutation process of DELA guides the FLA to move towards the global optimum.

FLA optimization is formulated by the localization function as given by the Eq. 2. Brightest firefly is the most optimal estimated location information of the sensor node. These brightest fireflies are mutated to get better estimated location information of the sensor nodes. The ability of differential evolution to optimize non differential, non linear and non continuous real life problem is used in this hybrid algorithm for mutating firefly.

HFDLA evaluates the light intensity or brightness for each firefly in the population. The localization function value is calculated to find the brightness of the firefly. HFDLA algorithm makes the firefly to move towards the brighter firefly in the population. The distance moved and brightness of each firefly is calculated and then the best firefly position, which is the optimal location information of the sensor node, is estimated.

The design steps for hybrid firefly differential evolution localization algorithm are as follows.

Step 1: Firefly is a set of control variables of sensor node localization function
Step 2: Initialize fireflies' population size of 20 within the solution space
Step 3: The localization objective function is calculated to find the brightness of the fireflies using the Eq. 2
Step 4: Attractiveness of firefly towards other fireflies is calculated using the Eq. 6
Step 5: Distance between the fireflies calculated using the Eq. 7
Step 6: Firefly i move towards brighter j firefly using the Eq. 8
Step 7: Hybrid firefly differential evolution localization algorithm uses differential evolution algorithms mutation process using the Eq. 3. The objective of mutation is to enable search diversity in the parameter space and provide suitable parameter variation to achieve better optimal estimate of location information of the sensor nodes
Step 8: Rank the fireflies and find the current global best

Step 9: Continue to execute the steps 4, 5, 6, 7 and 8 till the stopping criterion of 100 iterations is satisfied

Step 10: Estimate the optimal sensor node location information

The output of the hybrid firefly differential evolution localization algorithm is shown in the Fig. 5.

The convergence curve of the hybrid firefly differential evolution localization algorithm is shown in the Fig. 6.

Table 3 provides the details about the parameters considered in the design of hybrid firefly differential evolution localization algorithm.

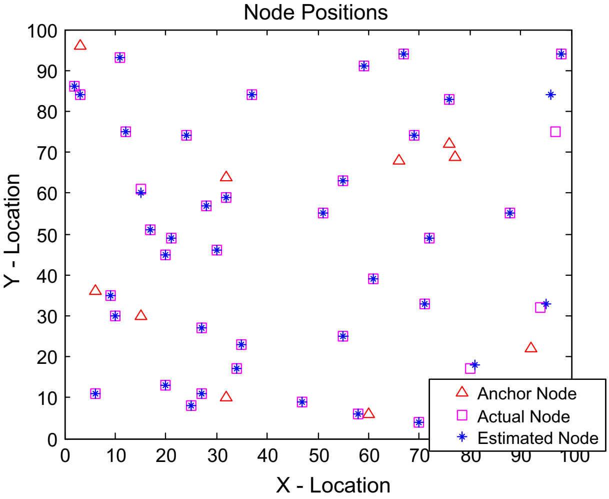

Fig. 5 Output of the hybrid firefly differential evolution localization algorithm

Fig. 6 Convergence curve of the hybrid firefly differential evolution localization algorithm

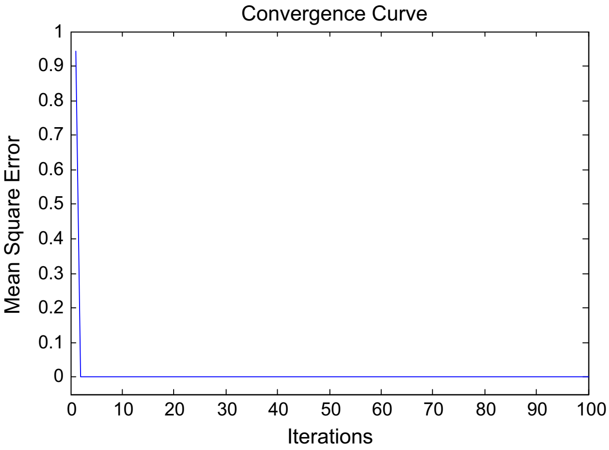

Table 3 Design parameters of hybrid firefly differential evolution localization algorithm

Parameters	Values
Maximum iteration	100
Space size	99
Landmarks	10
Unknown node	40
Total node	50
Number of fireflies	20
Randomizing coefficient, α	0.2
Initial attractiveness at $d = 0$, A_o	1.0
Absorption coefficient, γ	0.96
Scaling factor, S	0.7
Cross over constant, CC	0.6

Table 4 The output analysis of the three algorithms FLA, DELA, HFDLA

S. no.	Localization algorithm	Computation time (s)	Accuracy (%)	Convergence (iteration)
1	DELA	0.457125	78	Twelfth
2	FLA	0.765107	94	Tenth
3	HFDLA	1.051185	98	Third

6 Result and Analysis

The location information of the sensor nodes is estimated using three nature inspired localization algorithms namely differential evolution localization algorithm (DELA), firefly localization algorithm (FLA), and hybrid firefly differential evolution localization algorithm (HFDLA). The Table 4 gives the comparative analysis for the localization problem scenario as discussed in Sect. 2.

From the Table 4 it can be concluded that the accuracy of hybrid firefly differential evolution localization algorithm is better, has better convergence compared to differential evolution localization algorithm and firefly localization algorithm. But the computation time of hybrid firefly differential evolution localization algorithm is higher compared to other two algorithms.

7 Conclusion

In this paper three nature inspired intelligent algorithms namely differential evolution localization algorithm, firefly localization algorithm, and hybrid firefly differential evolution localization algorithm are proposed, analyzed, implemented and compared. The algorithms uses range based distributed localization algorithm. Therefore the communication of location information over several sensor nodes to the sink node is avoided in a multi hop environment. This increases the lifetime, reliability and performance of the wireless sensor networks. Out of the three algorithms hybrid firefly differential evolution localization algorithm has better accuracy and convergence compared to differential evolution localization algorithm and firefly localization algorithm, but takes more computation time comparatively. The output performance of the algorithms can be improved for better accuracy, convergence and computation time by varying the design parameters. Different hybrid mechanisms for algorithms can be analyzed for better optimal solution for localization in wireless sensor networks.

References

1. Mao, G., Fidan, B., & Anderson, B. D. O. (2007). Wireless sensor network localization techniques. *Science Direct, Computer Networks, 51*, 2529–2553.
2. Han, G., Xu, H., Duong, T. Q., Jiang, J., & Hara, T. (2011). Localization algorithms of wireless sensor networks: A survey. *Telecommunication Systems, 52*(4), 2419–2436.
3. Karl, H., & Willig, A. (2005). *Protocols and architectures for wireless sensor networks*. New York: Wiley.
4. Kulkarni, R. V., Förster, A., & Venayagamoorthy, G. K. (2011). Computational intelligence in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials, 13*(1), 68–96.
5. Ren, X., Gao, C., & Xi, Y. (2013). A node localization algorithm based on simple particle swarm optimization in wireless sensor networks. *Journal of Computational Information Systems, 9*(22), 9203–9210.
6. Guo, H., Low, K.-S., & Nguyen, H.-A. (2011). Optimizing the localization of a wireless sensor network in real time based on a low-cost microcontroller. *IEEE Transactions on Industrial Electronics, 58*(3), 741–749.
7. Mao, G., & Fidan, B. (2009). *Localization algorithms and strategies for wireless sensor networks: Monitoring and surveillance techniques for target tracking* (pp. 1–32). Hershey, PA: IGI Global.
8. Salman, N., Ghogho, M., & Kemp, A. H. (2014). Optimized low complexity sensor node positioning in wireless sensor networks. *IEEE Sensors Journal, 14*(1), 39–46.
9. Shi, Q., He, C., Chen, H., & Jiang, L. (2010). Distributed wireless sensor network localization via sequential greedy optimization algorithm. *IEEE Transactions on Signal Processing, 58*(6), 3328–3340.
10. Brownlee, J. (2011). *Clever algorithms: Nature-inspired programming recipes.*
11. Kulkarni, R. V., & Venayagamoorthy, G. K. (2011). Particle swarm optimization in wireless sensor networks: A brief survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 41*(2), 262–267.
12. Binitha, S., & Sathya, S. S. (2012). A survey of bio inspired optimization algorithms. *International Journal of Soft Computing and Engineering (IJSCE), 2*(2), 137–151; ISSN: 2231-2307.
13. Storn, R., & Price, K. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*, 341–359.
14. Vaisakh, K., & Srinivas, L. R. (2008). Differential evolution approach for optimal power flow solution. *Journal of Theoretical and Applied Information Technology, 4*(4), 261–268.
15. Yang, X. S. (2008). *Nature-inspired meta-heuristic algorithms*. Beckington: Luniver Press.
16. Apostolopoulos, T., & Vlachos, A. (2011). Application of the firefly algorithm for solving the economic emissions load dispatch problem. *Hindawi Publishing Corporation International Journal of Combinatorics.* doi:10.1155/2011/523806.
17. Yang, X.-S., Hosseini, S. S. S., & Gandomic, A. H. (2012). Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. *Elsevier Applied Soft Computing, 12*, 1180–1186.

R. Harikrishnan is currently Ph.D. candidate at Faculty of Electrical and Electronics Engineering, Sathyabama University, Chennai, India. His areas of interests are Wireless Sensor Networks, Computational Intelligence, Vehicular Networks and Embedded Systems.

V. Jawahar Senthil Kumar received his Ph.D. degree from college of Engineering Guindy, Anna University, Chennai, India. He is currently an Associate Professor at Department of Electronics and Communication Engineering, CEG, Anna University, Chennai, India. His areas of interest include Wireless Sensor Networks, Computer Networks and VLSI.

P. Sridevi Ponmalar is currently Ph.D. candidate at Faculty of Information and Communication, College of Engineering Guindy, Chennai, India. Her areas of Interest are Wireless Sensor Networks, Soft Computing and Embedded Systems.