Hybrid Genetic Algorithm–Differential Evolution Approach for Localization in WSN

P. Srideviponmalar, V. Jawahar Senthil Kumar and R. Harikrishnan

Abstract Nature-inspired algorithms have the characteristics to learn and decide and to be adaptable, intelligent, and robust, and so they can be used for solving complex problems. This paper deals with one such algorithm named hybrid genetic algorithm–differential evolution for localization in wireless sensor network. This algorithm is used to estimate the position of sensor node. A novel hybrid algorithm is analyzed, designed, and implemented. This algorithm provides better accuracy and is simple to implement.

Keywords Genetic algorithm · Wireless sensor network · Sensor node
Anchor node · Differential evolution

1 Introduction

Internet of things and other intelligent monitoring techniques need sensors to be embedded with the devices under control. The sensor node localization would reduce the signal traffic which otherwise can lead to huge traffic of signals for location estimation and location update [1]. In most of the applications, the sensed information becomes meaningful only with the addition of location parameter [2].

P. Srideviponmalar (✉)
Faculty of Information and Communication Engineering, Anna University, Chennai, India
e-mail: sridevi_ponmalar@yahoo.co.in

V. Jawahar Senthil Kumar
Department of Electronics and Communication Engineering,
Anna University, Chennai, India
e-mail: veerajawahar@gmail.com

R. Harikrishnan
School of Electrical and Electronics Engineering,
Sathyabama University, Chennai, India
e-mail: rhareish@gmail.com

V. Bhateja et al. (eds.), *Intelligent Engineering Informatics*, Advances in Intelligent Systems and Computing 695, https://doi.org/10.1007/978-981-10-7566-7_27

Depending upon the application, the sensor nodes are connected in a topology to form wireless sensor networks (WSNs). The WSNs are used for applications such as forest fire detection, road traffic management, healthcare monitoring, animal habitat monitoring, precision agriculture, disaster management, military surveillance, and environmental monitoring [3].

The locations of sensor nodes can be predicted by analytical method. These methods are complex, and process becomes tedious with scalability of the network. So for this type of complex problems, nature-inspired algorithms are discussed in the literature. Because of the intelligence, robustness, flexibility, and simple implementation, nature-inspired algorithm is preferred [4].

Global positioning system (GPS) can be used for sensor node position detection. But GPS technique would consume more power from energy-constraint WSN. Moreover, every node has to connect with GPS which is costly [5]. This paper deals with a nature-inspired hybrid algorithm known as hybrid genetic algorithm–differential evolution localization algorithm (GADELA). This is a range-based distributed algorithm. A centralized algorithm requires sensor information to be transmitted to a centralized process. Therefore, the energy required compared to distributed algorithm is more. In distributed algorithm, the sensor location information need not be transmitted to a centralized process. Thus, the algorithm is energy efficient as the process is done locally [6]. In range-based algorithm, the distance is calculated by using radio signal strength indicator. The need for any extra hardware is not required for calculating the distance which is required for location estimation technique [7].

2 Localization Problem Modeling

A total of 50 nodes are taken within the solution space, in which 10 are the landmark nodes for which the location information in known priori and the remaining 40 are sensor nodes for which the location has to be found. Using trilateration method, the location of 40 sensor nodes is estimated by using the 10 landmark nodes as shown in Fig. 1.

This type of estimation follows two steps: First step is the distance calculation, and the next one is the position estimation. The distance calculation is carried out by using the Eq. (1):

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

The position estimation is carried out by using Eq. (2):

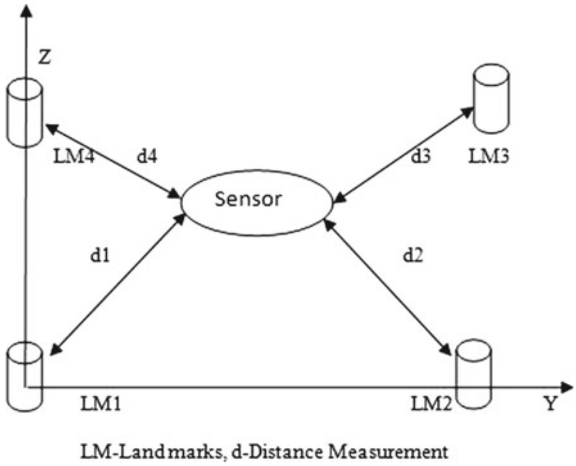


Fig. 1 Trilateration method

$$f(y_u, z_u) = \left[\sqrt{(y_{est} - y_{landmark})^2 + (z_{est} - z_{landmark})^2} - d_{ist}\right]^2 \tag{2}$$

The average position estimation is calculated by using Eq. (3):

$$f(y_u, z_u) = \left[\sqrt{(y_{est} - y_{landmark})^2 + (z_{est} - z_{landmark})^2} - d_{ist}\right]^2 / N \tag{3}$$

y_{act}, z_{act} is the actual y and z positions of sensor node, respectively.
y_{est}, z_{est} is the estimated y and z positions, respectively, of the sensor node which does not know the location.
$y_{landmark}, z_{landmark}$ is the anchor or landmark of y and z positions, respectively.
N is the total sensor nodes.

3 Hybrid Genetic Algorithm–Differential Evolution (GADE) Localization Algorithm

Genetic algorithm (GA) solves nonlinear, nonconvex operation problems. It uses the nature's concept of survival of fittest. For the localization of sensor node problem, four stages of optimization are carried out. The localization error is optimized, so that optimal localization information is estimated. The four stages are initialization, selection, crossover, and mutation. Initialization of sensor node population is carried out first. Genetic algorithm performs better selection

Table 1 Hybrid GADELA design parameters

Parameters (localization)	Tuned values
Maximum no. of iterations	100
Space size	100
No. of landmarks	10
No. of unknown node	40
No. of total node	50
No. of vector	20
Crossover constant (GA)	0.3
Scaling factor, S (DE)	0.9

operation and crossover operation. So these operations are carried out by genetic algorithm [8].

Differential evolution performs better mutation. So the mutation operation is carried out by differential evolution (DE). DE has the ability to solve non-continuous and non-differential real-world problems. DE performs better on mutation operation [9]. It mutates vector with other vectors within the given population which are randomly selected. Mutation operation leads to global optimization. So hybrid genetic algorithm and differential evolution would give better optimal location of sensor nodes. So this idea is implemented in this GA-based DE algorithm.

Following is the design procedure for hybrid genetic algorithm–differential evolution localization algorithm:

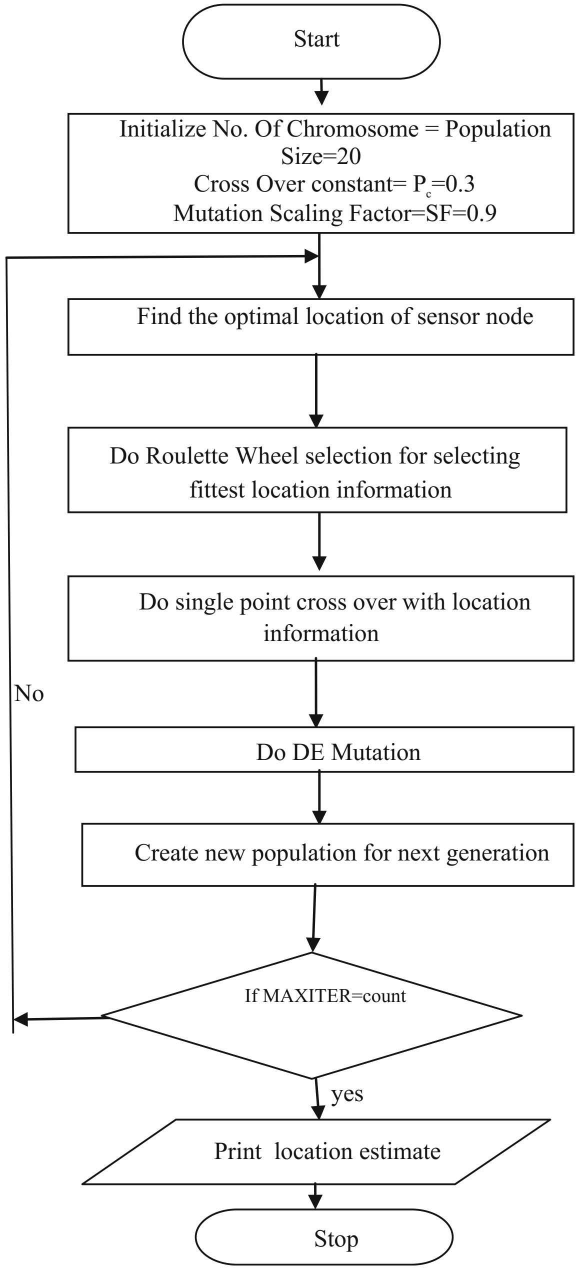
Step 1 Control variables of localization of sensor nodes are selected as genes of a chromosome.
Step 2 Create initial population of sensor nodes.
Step 3 Find fitness of sensor node location information (chromosomes) by using the localization function.
Step 4 Select location information using Roulette wheel selection for mating.
Step 5 Perform crossover operation as in genetic algorithm.
Step 6 Perform mutation operation as in differential evolution.
Step 7 Select new population of location of sensor nodes for the next generation.
Step 8 Repeat steps 4, 5, 6, and 7 for stopping the condition.
Step 9 Print location estimate.

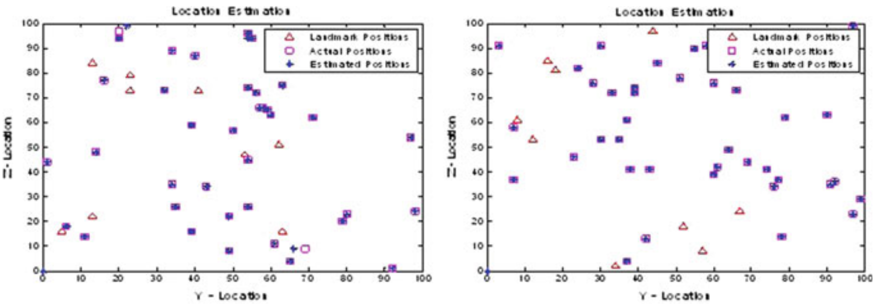
Design consideration for hybrid GADELA is recorded in Table 1. The flowchart of hybrid GADELA is shown in Fig. 2.

4 Result Analysis

Hybrid GADELA is used to estimate the location. The outputs of hybrid GADE localization algorithm are shown in Figs. 3, 4, 5, 6, and 7.

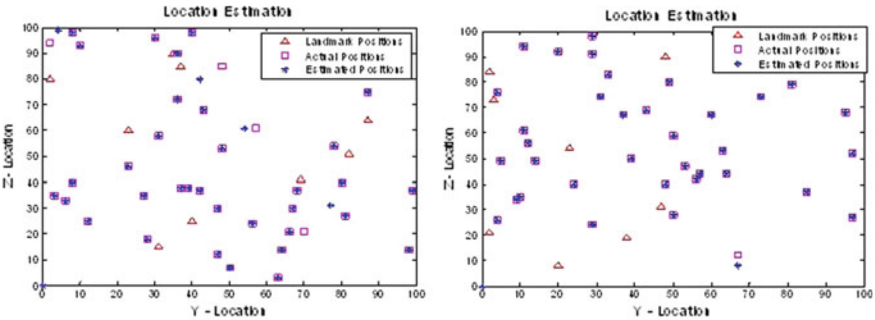
Fig. 2 Flowchart of hybrid GADE localization algorithm





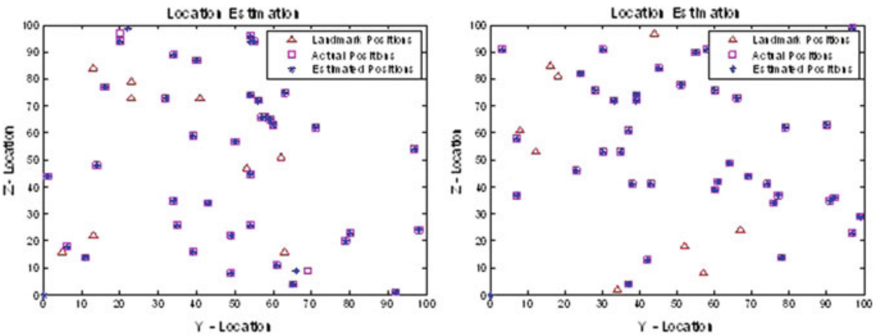
(a) Localization Error (b)Average localization Error

Fig. 3 Output of GADELA for 100 population vector size



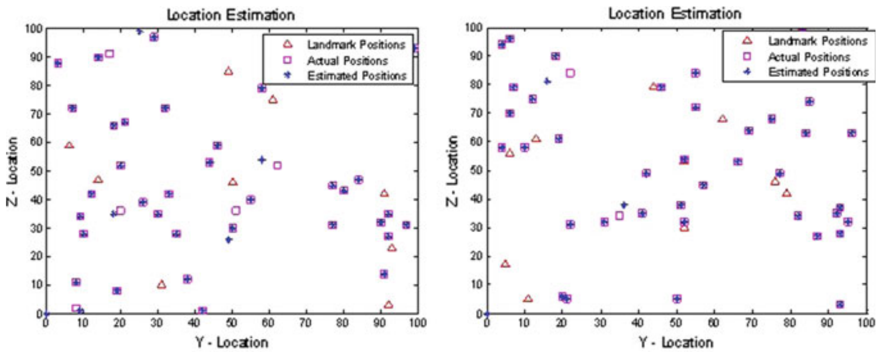
(a)Localization Error (b)Average localization Error

Fig. 4 Output of GADELA for 80 population vector size



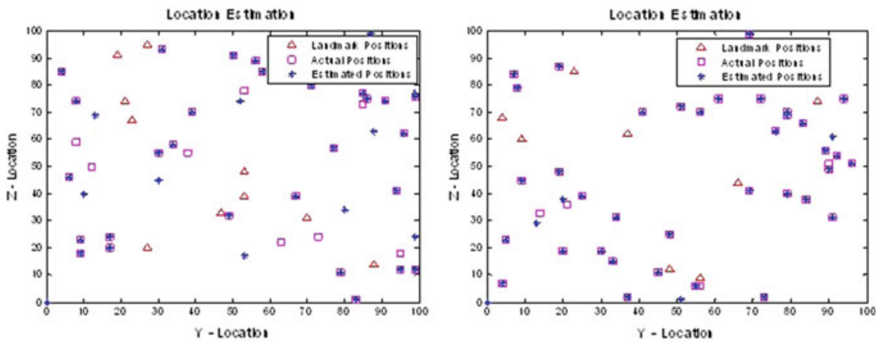
(a) Localization Error (b)Average localization Error

Fig. 5 Output of GADELA for 60 population vector size



(a)Localization Error (b)Average localization Error

Fig. 6 Output of GADELA for 40 population vector size



(a)Localization Error (b)Average localization Error

Fig. 7 Output of GADELA for 20 population vector size

The output of GADELA in terms of location error and average localization error is shown in Figs. 3, 4, 5, 6 and 7 for a population vector size of 100, 80, 60, 40 and 20 respectively.

The performance analysis of hybrid GADE localization algorithm with localization function is presented in Table 2. It presents the accuracy and time complexity of the hybrid algorithm for various population vector sizes under consideration with localization function. Similarly, Table 3 shows the accuracy and time complexity of the hybrid algorithm for various population vector sizes under consideration with average localization function. By comparison, it is found that the accuracy increases and the time complexity performance is better with the increase in population vector size. And also the hybrid GADE localization algorithm performs better with accuracy and time complexity if it uses average localization function instead of simple localization function.

Table 2 Performance analysis of hybrid GADE localization algorithm with localization function

No. of population vector	Hybrid GADE localization algorithm with localization function	
	Accuracy	Time complexity
20	78	28.456
40	82	26.456
60	84	17.678
80	90	12.438
100	94	10.002

Table 3 Performance analysis of hybrid GADE localization algorithm with average localization function

No. of population vector	Hybrid GADE localization algorithm with average localization function	
	Accuracy	Time complexity
20	88	15.564
40	94	14.987
60	96	12.845
80	98	10.765
100	100	8.789

5 Conclusion

The hybrid GADELA is analyzed, designed, and implemented. In general, the algorithm shows better accuracy and it has better time complexity. The algorithm performs well in terms of accuracy and time complexity as the population vector size is increased. Further by using average localization function, the performance is further enhanced to get better accuracy and time complexity. The performance comparisons are provided. A careful selection of design parameters and better hybrid techniques might lead to improved performance and estimation accuracy.

References

1. D'Ore, S., Galluccio, L., Morabito, G., Palazzo, S.: Exploiting object group localization in the internet of things: performance analysis. IEEE Trans. Veh. Technol. **64**(8), 3645–3656 (2015)
2. Singh, S.P., Sharma, S.C.: Range free localization techniques in wireless sensor network: a review. Procedia Comput. Sci. **57**, 7–16 (2015). Science Direct
3. Mao, G., Fidan, B., Anderson, B.D.: Wireless sensor network localization techniques. Comput. Netw. **51**, 2529–2553 (2007). Science Direct
4. Raghavendra Kulkarni, V., Förster, A., Venayagamoorthy, G.K.: Computational intelligence in wireless sensor networks: a survey. IEEE Commun. Surv. Tutor. **13**(1), 68–96 (2011)
5. Halders, S., Ghosal, A.: A survey on mobility-assisted localization techniques in wireless sensor networks. J. Netw. Comput. Appl. 82–94 (2016)

6. Shi, Q., He, C., Chen, H., Jiang, L.: Distributed wireless sensor network localization via sequential greedy optimization algorithm. IEEE Trans. Signal Process. **58**(6), 3328–3340 (2010)
7. Salman, N., Ghogho, M., Kemp, A.H.: Optimized low complexity sensor node positioning in wireless sensor networks. IEEE Sens. J. **14**(1), 39–46 (2014)
8. Grefenstette, J.J.: Optimization of control parameters for genetic algorithms. IEEE Trans. Syst. Man Cybern. **16**(1), 122–128 (1986)
9. Vaisakh, K., Srinivas, L.R.: Differential evolution approach for optimal power flow solution. J. Theor. Appl. Inform. Technol. 261–268 (2008)