Evolutionary Intelligence (2019) 12:469–478
https://doi.org/10.1007/s12065-019-00239-0

RESEARCH PAPER

Implementation of self adaptive mutation factor and cross-over probability based differential evolution algorithm for node localization in wireless sensor networks

Visalakshi Annepu[1] · A. Rajesh[1]

Received: 1 November 2018 / Revised: 29 March 2019 / Accepted: 2 May 2019 / Published online: 14 May 2019

Abstract
Node localization or positioning is essential for many position aware protocols in a wireless sensor network. The classical global poisoning system used for node localization is limited because of its high cost and its unavailability in the indoor environments. So, several localization algorithms have been proposed in the recent past to improve localization accuracy and to reduce implementation cost. One of the popular approaches of localization is to define localization as a least square localization (LSL) problem. During optimization of LSL problem, the performance of the classical Gauss–Newton method is limited because it can be trapped by local minima. By contrast, differential evolution (DE) algorithm has high localization accuracy because it has an ability to determine global optimal solution to the LSL problem. However, the convergence speed of the conventional DE algorithm is low as it uses fixed values of mutation factor and cross-over probability. Thus, in this paper, a self-adaptive mutation factor cross-over probability based differential evolution (SA-MCDE) algorithm is proposed for LSL problem to improve convergence speed. The SA-MCDE algorithm adaptively adjusts the mutation factor and cross-over probability in each generation to better explore and exploit the global optimal solution. Thus, improved localization accuracy with high convergence speed is expected from the SA-MCDE algorithm. The rigorous simulation results conducted for several localization algorithms declare that the propose SA-MCDE based localization has about (40–90) % more localization accuracy over the classical techniques.

Keywords Localizations · Wireless sensor networks · Gauss–Newton · DE algorithm · SA-MCDE algorithm

1 Introduction

The use of wireless sensor network (WSN) has become quite predominant in recent years because of availability of low cost and energy efficient wireless sensor nodes [1]. Large number of such sensor nodes can be connected to form a WSN that can be used for disaster relief operation, military surveillance, environmental monitoring, medical observations, home applications and so on [2]. The sensor nodes in WSN can be deployed either randomly or in a systematic way. During sensing certain phenomena (such as temperature, humidity, pressure, position, vibration, sound, chemical concentration etc.), each node in a WSN is supposed to know its location in physical world. If the location of the event is unknown, detection of that event is not particularly useful [3].

The global poisoning system (GPS) is a popular scheme that is used for outdoor positioning and this scheme requires a clear line of sight (LOS) between the satellite and the unknown node without any obstacles [4]. However, the GPS has limitations, such as high energy consumption and expensive when it is deployed in a large quantity. Hence, it is not recommended to deploy GPS module in each node. Alternatively, few nodes can be selected as beacons (or anchors) and GPS modules can be deployed only in these beacons. Through localization process and with the aid of these beacons, remaining nodes can determine their locations. This strategy can effectively minimize energy consumption as well as the cost of implementation.

✉ A. Rajesh
rajeshtechece@gmail.com

Visalakshi Annepu
annepuvisala@gmail.com

[1] School of Electronics Engineering (SENSE), Vellore Institute of Technology, Vellore, India

During last two decades, numerus localization techniques were introduced to minimize localization error. All these localization techniques are mainly classified as classical techniques and soft computing based techniques. The classical localization schemes are again classified as range based and range free schemes. Range-based localization techniques require distance or angle between sensor nodes to estimate node position. Using simple geometry, this information can be used to derive information about node positions. When distances between entities are used, the approach is called multilateration; when angles between nodes are used, one talks about angulation. Range-free localization algorithms do not require distance or angle information between target node and anchor node, but depends on the topological information.

Range-based algorithms provide better accuracy compared to range-free localization algorithms, but they are not so economic [5]. Examples of this range based schemes angle-of-arrival (AoA) [6], time of arrival (ToA) [7], time difference of arrival (TDoA) [8], acoustic energy [9], and received signal strength indicator (RSSI) [10, 11]. The AoA technique requires deploying an extra antenna in an appropriate direction, which may impose additional cost. Both the ToA and TDoA techniques require that all the receiving nodes to be perfectly synchronized. These techniques have high positioning accuracy but need additional hardware. Thus, among all range based techniques, the RSSI is cost effective as it does not required much equipment, but its accuracy still needs to be improved.

By contrast, the range-free techniques are based on pattern matching and Hop count but not on distance or angle. Thus their implementation cost is low at the price of low accuracy. Examples of this approach include centroid algorithm [12], DV-Hop technique [13], and amorphous algorithm [14]. In centroid algorithm, nodes infer proximity to reference points and then localize themselves to the centroid of the reference points. The DV-Hop used Hop count between nodes instead of distance estimation. Hence, when the DV-Hop is used for implementing in high dense sensor network, its poisoning error also increases consistently. Unlike DV-Hop, the amorphous algorithm computed Hop distance instead of linear distance among the nodes. Hence, it's positioning accuracy higher than DV-Hop algorithm. There is a clear trade-off between range based and range free techniques with respect to poisoning accuracy and implementation cost. Hence localization is defined as least square localization (LSL) problem and it is solved by several soft computing techniques.

Recently, the applications of soft computing techniques had gained a great interest. Since, the mobile node localization also can be formulated as an optimization problem, the gradient technique such as Gauss–Newton algorithm is used for location in [15]. However, the gradient based techniques have a possibility of trapped by local minima and also these techniques require appropriate initial conditions of desired parameters. Thus, the gradient based techniques can be replaced by metaheuristic optimization techniques (MOTs) to obtain better localization accuracy. Various MOTs such as flower pollination algorithm (FPA), firefly algorithm (FA), grey wolf optimization (GWO), particle swarm optimization (PSO), bacterial foraging algorithm (BFA), butterfly optimization algorithm (BOA), particle filtering NLS (PF-NLS) initial optimization algorithms have used for node localization of sensor networks in the literature [16–22]. Similarly, the popular differential evolution (DE) algorithm is also applied for node localization in the literature [23–25]. Though each MOT has its own merits, the DE algorithm proposed by Storn and Price [26] has become popular because of its ability to reach global optima irrespective of initial parameters and consists of limited control parameters. Despite DE algorithm has an excellent performance, its convergence speed still seems slow because the DE algorithm is sensitive to the choice of mutation factor and cross over probability. In general, selection of such control parameters is a tedious task. The fixed control parameter does not guarantee the better individuals. As a result, a self-adaptive mutation factor and cross-over probability differential evolution (SA-MCDE) algorithm was proposed, which has self-adaptive mechanism to vary mutation factor and cross over probability in each generation [27]. The SA-MCDE algorithm thoroughly explores solution in the initial generations and exploits more the solution in the final few generations. Thus, this adaptive selection of control parameters guarantees better solutions and also provides high convergence speed. So, this paper aims at designing SA-MCDE based localization technique, that has fast convergence, accurate positioning and low implementation cost.

The paper is organized as follows. Optimization of localization problem along with Gauss–Newton method is described in Sect. 2. The proposed SA-MCDE localization technique in detail is given in Sect. 3. The simulation analyses and conclusions are presented in Sects. 4 and 5 respectively.

2 Localization problem formulation

Let us consider, in a given two dimensional sensor field, N position aware anchor nodes a_n, $n = 1, 2, \ldots, N$ and M unknown sensor nodes b_m, $m = 1, 2, \ldots, M$ are deployed. Deployment of these nodes can be either systematic way or in a random fashion. The positions of nth anchor node and mth unknown node are given as (x_n, y_n) and (x_m, y_m) respectively. All these nodes are assumed to be with in the radio rang R, therefore $a_n, b_m \in R^2$. It is also assumed that, each unknown node have distance measurement from all anchor nodes. That means, the

mth node contains Euclidean distance d_{nm} between a_n and b_m. This distances d_{nm} is also assumed to be within the radio range R, because a communication link exists between a_n and b_m only if the distances d_{nm} is less than R. The aim of unknown node localization is to estimate the coordinates of M unknown nodes using N anchor nodes. Thus, localization problem can be formulated to determine coordinates of b_m with the obtained distance measurements d_{nm} as:

$$F(u_m) = \sum_n \left|s_{nm}(b_m)\right|^2 \tag{1}$$

$$s_{nm}(u) = \left\|a_n - b_m\right\|^2 - \hat{d}^2_{nm}, \quad \text{if} \left\|a_n - b_m\right\| \le R$$

where $\left\|a_n - b_m\right\| = \sum_{n=1}^{N} \sqrt{\left(x_n - x_m\right)^2 + \left(y_n - y_m\right)^2}$ and $\hat{d}_{nm}$ is the noisy measurement of the distance d_{nm}, which is obtained from RSSI. The noisy distance measurement is given by:

$$\hat{d}_{nm} = d_{nm} \times \text{ noise factor} \times (\text{rand}/2) \tag{2}$$

Hence, the solution from localization problem can be obtained from:

$$b_m = \arg \min_{b_m} F(b_m) \tag{3}$$

A centralized Gauss–Newton method is commonly used for solving this least squares localization problem. The Gauss–Newton technique is a gradient descent technique that updates the unknown node position in the reverse direction of their gradient. The unknown node location is updated in the $(i+1)$th iteration as:

$$b_m(i+1) = b_m(i) + \alpha p_m(i) \tag{4}$$

Let, $A = \nabla_{b_m}\left[s_{nm}(b_m)\right]^T = \frac{\partial}{\partial b_m} F(b_m)$ and $B = \left[\hat{d}_{nm}\right]^T$ then

$$p_m(i) = -\left(A^T A\right)^{-1} A^T B \tag{5}$$

Here, α is rate learning parameter.

The gradient based methods such as Gauss–Newton has high converge speed, but they can be trapped by local minima. Also, gradient techniques need appropriate choice of initial states along with differentiable and continuous cost functions. These difficulties may be overcome by replacing gradient descent techniques with MOTs [12]. The MOTs work without derivative information of the cost function and hence can accept discontinuous cost functions. The MOTs update the desired parameters from random initial positions and converge the cost function precisely to global minima.

3 Proposed SA-MCDE algorithm based localization

Among the various optimization techniques, the DE algorithm is found to be more appropriate due to its fast convergence and optimal solutions [26]. However, the DE algorithm is much sensitive to selection of mutation factor and cross-over probability. So, and SA-MCDE was proposed, which adapts mutation factor and cross-over probability in each generation [27]. The adaptive mutation factor may retain population diversity and adaptive cross-over probability may firmly explore the search space for global optima. So, an improved solution is expected from SA-MCDE, a thus it has been applied in several engineering applications [28–30]. SA-MCDE is a simple, efficient and robust optimization technique. The SA-MCDE algorithm uses four stages, namely, initialization, mutation, crossover and selection to obtain global optimum. Figure 1 shows the flow chart of the SA-MCDE employed for the node localization. The process implemented for node localization is summarized as follows.

Initialization SA-MCDE begins its search from a randomly initialized population containing P individuals, where each individual contains two variables corresponding to x and y axes of unknown node. The pth individual of the $(P \times 2)$—dimensional population set $\boldsymbol{w}_m$ in the gth generation is expressed as:

$$\boldsymbol{w}_m^{g,p} = \left[x_m^{g,p}, y_m^{g,p}\right]^T, \; p = 1, 2, \ldots, P, \;\; g = 1, 2, \ldots, G \tag{6}$$

where m refers to index of unknown node.

Adaptive Mutation and crossover Mutation operation explores the search space with an adaptive mutation factor F. In each generation, DE algorithm randomly samples three other individuals $\boldsymbol{w}_m^{g,r_1}$, $\boldsymbol{w}_m^{g,r_2}$ and $\boldsymbol{w}_m^{g,r_2}$ from the same generation and produces a mutant vector $v_m^{g,p}$ by adding an appropriately scaled difference vector of two individual to the third vector. The indexes of these randomly chosen vectors are integers, mutually different and r_1, r_2, $r_3 \in \{1, 2, \ldots, P\}$. The randomly chosen integers r_1, r_2 and r_3 are also taken to be different from the running index p. This can be expressed mathematically as follows:

$$\boldsymbol{v}_m^{g,p} = \boldsymbol{w}_m^{g,r_1} + F\left(\boldsymbol{w}_m^{g,r_2} - \boldsymbol{w}_m^{g,r_3}\right) \tag{7}$$

The mutation factor $F \in (0, 1]$ is a positive real-valued number and its controls the amplification of the differential variation $\left(\boldsymbol{w}_m^{g,r_2} - \boldsymbol{w}_m^{g,r_3}\right)$.

In order to improve diversity of the perturbed vectors, crossover operation is used. The crossover operation generates a trail vector $\boldsymbol{u}_m$ by restoring certain parameters of the target vector $\boldsymbol{w}_m$ with randomly selected parameters of mutant vector $\boldsymbol{v}_m$. The crossover operator effectively shuffles parameters with a crossover probability $C_p \in$(0,

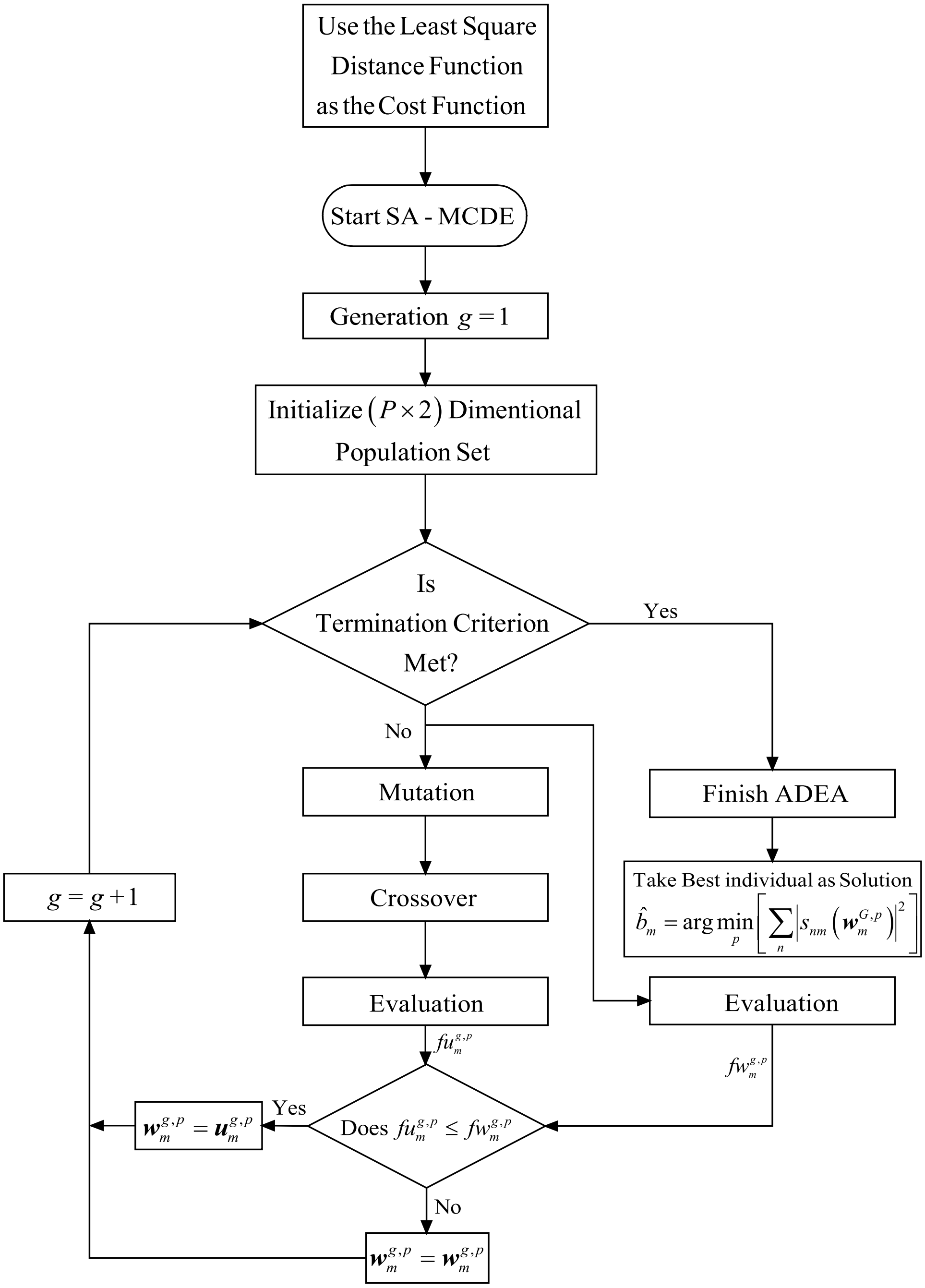
Use the Least Square Distance Function as the Cost Function
Start SA - MCDE
Generation $g = 1$
Initialize $(P \times 2)$ Dimentional Population Set
Is Termination Criterion Met?
Yes
No
Mutation
Crossover
Evaluation
$fu_m^{g,p}$
Finish ADEA
Take Best individual as Solution
$\hat{b}_m = \arg\min_p \left[\sum_n \left| s_{nm} \left(\boldsymbol{w}_m^{G,p} \right) \right|^2 \right]$
Evaluation
$fw_m^{g,p}$
Does $fu_m^{g,p} \le fw_m^{g,p}$
Yes
No
$\boldsymbol{w}_m^{g,p} = \boldsymbol{u}_m^{g,p}$
$\boldsymbol{w}_m^{g,p} = \boldsymbol{w}_m^{g,p}$
$g = g + 1$

Fig. 1 Flowchart of SA-MCDE based localization algorithm

1] about to increase the search space for a better solution. The obtained dth variable in the trail vector is given by:

$$\boldsymbol{u}_m^{g,p,d} = \begin{cases} \boldsymbol{v}_m^{g,p,d} & \text{rand}_d(0,1) \le C_p \text{ or } d = d_{rand}, \\ \boldsymbol{w}_m^{g,p,d} & \text{otherwise.} \end{cases} \tag{8}$$

where $d = 1$ or 2, that means 1 corresponding to x-axis position and 2 corresponding to y-axis position. Further, $\text{rand}_d \in [0, 1]$ is the random value generated for dth variable and d_{rand} is a randomly chosen index, that is 1 or 2, which ensures that the trail vector gets at least one parameter from mutant vector.

The SADA algorithm uses self-adaptive control strategy for varying the mutation factor F and crossover probability C_p during each generation in order to find best solution with high rate of convergence. The self-adaptive control parameters F^{g+1} and C_p^{g+1} are obtained in generation $g + 1$ as:

$$F^{g+1} = \begin{cases} F_1 + \text{rand}_1 F_2, & \text{if} \, \text{rand}_2 < \tau_1, \\ F^g, & \text{otherwise} \end{cases} \tag{9}$$

$$C_p^{g+1} = \begin{cases} \text{rand}_3, & \text{if rand}_4 < \tau_2, \\ C_p^g, & \text{otherwise} \end{cases} \quad (10)$$

where $\text{rand}_j \in \{1, 2, 3, 4\}$, are uniform pseudo-random values lies between 0 and 1, τ_1 and τ_2 are constants that are used to define F and C_p respectively. The values of τ_1, τ_2, F_1 and F_2 are taken to be 0.1, 0.1, 0.1 and 0.9, respectively according to [27]. The new F takes value from [0.1, 1.0] and the new C_p from [0, 1] randomly. The control parameter values F^{g+1} and C_p^{g+1} are updated prior to mutation as they involved in mutation, crossover, and selection operations.

Fitness evaluation In the fitness evaluation stage the trail vector $\boldsymbol{u}_m^{g,p}$ and the target vector $\boldsymbol{w}_m^{g,p}$ are fed to a fitness (cost) function to know their fitness value. The fitness values of the pth individual in the gth generation has been computed for both trail and target vectors from:

$$fu_m^{g,p} = \sum_n \left| s_{nm}(u_m^{g,p}) \right|^2 \quad (11)$$

$$fw_m^{g,p} = \sum_n \left| s_{nm}(w_m^{g,p}) \right|^2 \quad (12)$$

Selection The DE algorithm uses a greedy selection. The selection operation compares fitness value of each individual in both trail vectors and target vectors. The individual with minimum fitness value can be survived and carried to the next generation. So, the cost minimization problem ca be defined as:

$$\boldsymbol{w}_m^{g+1,p} = \begin{cases} \boldsymbol{u}_m^{g,p} & fu_m^{g,p} \le fw_m^{g,p}, \\ \boldsymbol{w}_m^{g,p} & \text{otherwise} \end{cases} \quad (13)$$

The whole procedure will be repeated from mutation stage till termination criterion is met. Termination can be defined either based on number of generations or meeting minimum acceptable fitness value.

Optimal solution The best individual with minimum fitness value is considered as the optimal position for unknown node m at the end of termination. From this optimal solution, the position of unknown node is estimated as:

$$\hat{b}_m = \arg\min_p \left[\sum_n \left| s_{nm}\left(\boldsymbol{w}_m^{G,p}\right) \right|^2 \right], \quad p = 1, 2, \ldots, P \quad (14)$$

Similarly, the entire procedure is repeated for finding the optimal positions of remaining $M-1$ nodes.

4 Simulation analyses and results

In this section, the computer simulations have been conducted to show effective ness of the proposed SA-MCDE Localization technique. The proposed and classical algorithms are simulated in Matlab. In these simulations, we have deployed 100 unknown sensor nodes and 10 position aware anchor nodes in a region of 100 × 100 square meters. Among 10 anchor nodes, 4 anchor nodes are placed at (25 m, 25 m), (25 m, 75 m), (75 m, 25 m), (75 m, 75 m), and remaining 6 anchor nodes are deployed randomly to increase connectivity in the network. All unknown nodes are placed at random positions. To eliminate randomness of position estimates, we have averaged the position estimates by applying each localization techniques 100 times repeatedly at each unknown node. The parameters of various localization techniques have been chosen based on extensive simulations conducted with different values. The selected parameters are listed in Table 1. The simulation results presented in this section mainly focusses on localization accuracy and convergence speed of various localization techniques. The localization accuracy can be computed mainly based on the measures namely, Localization Error (LE), Average Localization Error (ALE) and Root Mean Square Error (RMSE), which are defined in Eqs. (15), (16) and (17) respectively.

$$LE_m = \sqrt{(\hat{x}_m - x_m)^2 + (\hat{y}_m - y_m)^2} \quad (15)$$

$$ALE = \frac{1}{M} \sum_{m=1}^{M} \sqrt{(\hat{x}_m - x_m)^2 + (\hat{y}_m - y_m)^2} \quad (16)$$

$$RMSE = \sqrt{\frac{1}{M} \sum_{m=1}^{M} (\hat{x}_m - x_m)^2 + (\hat{y}_m - y_m)^2} \quad (17)$$

where M is number of unknown nodes, (x_m, y_m) is the true location of unknown node m, and $(\hat{x}_m, \hat{y}_m)$ is the estimated location of unknown node m.

Table 1 Simulation parameters

Parameter	Value
Gauss–Newton method	
Learning rate parameter ‘α’	0.2
Number of iterations	100
Differential evolution	
Number of generation	50
Population size	100
Mutation factor	0.5
Cross-over probability	0.6
SA-MCDE	
Number of generation	50
Population size	100
Mutation factor	Variable
Cross-over probability	Variable

Figure 2 shows distribution of anchor nodes, unknown nodes and estimated positions of unknown node using various localization techniques. From this figure, it is found that the localization accuracy of the proposed SA-MCDE localization is high compared to the classical technique. Among various localization schemes, the range free technique such as DV-Hop results high localization error. So, there is a huge deviation between true positions and estimated positions. By contrast, the range based techniques are fairly performing better compared to range DV-Hop technique. Among various

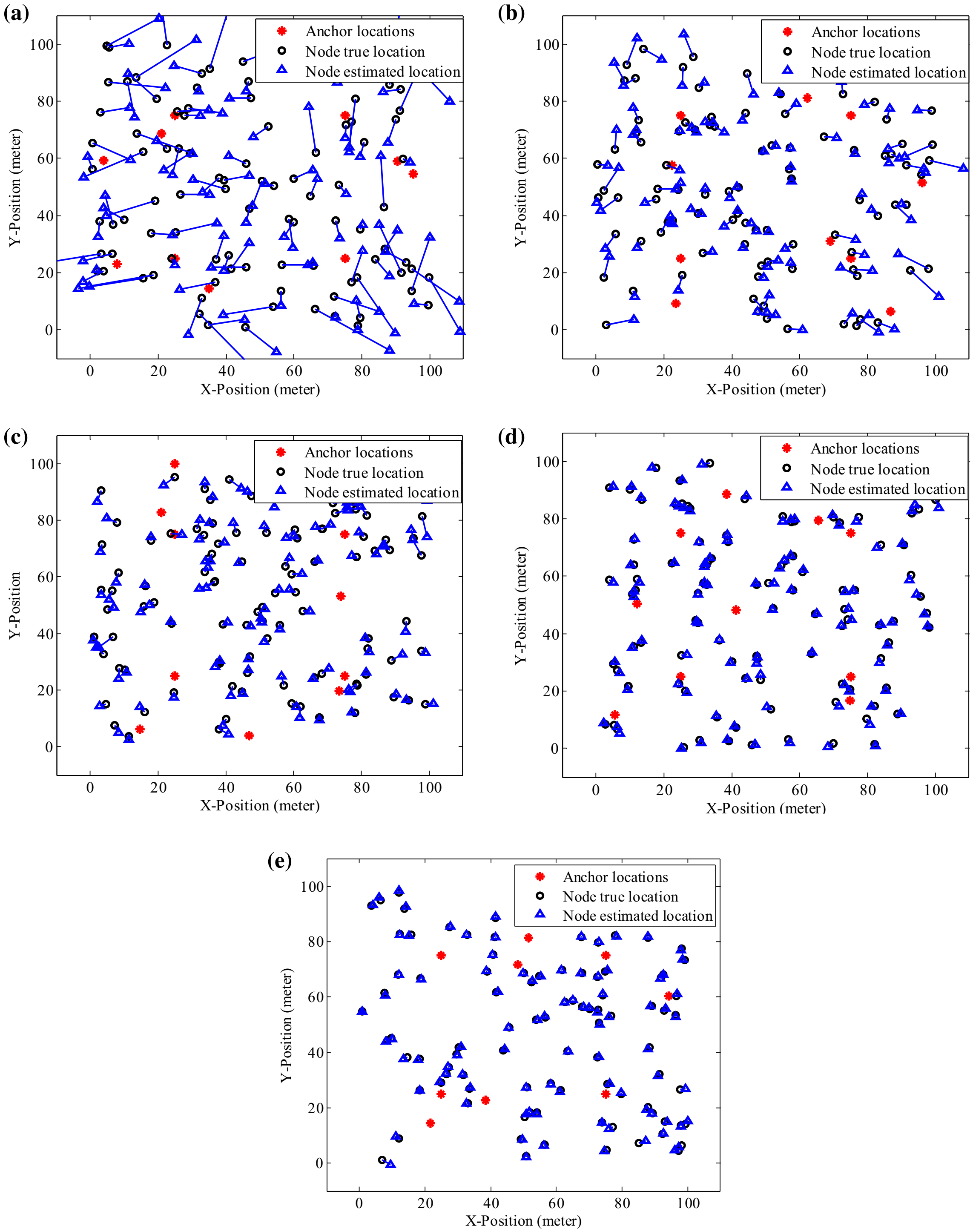

Fig. 2 Deployment anchor and unknown nodes along with estimated location of unknown nodes using: **a** DV-Hop, **b** RSSI, **c** Gauss–Newton, **d** DE algorithm, **e** SA-MCDE algorithm

range-based techniques, the proposed SA-MCDE localization has high degree of accuracy, as the true node position and estimated node position almost coincides, which is show in Fig. 2e. This is due to the fact that the SA-MCDE has true global optimization ability. Further, the localization error computed for each node in the network using different localization techniques is plotted in Fig. 3. As seen in this figure, there is a large variation of localization error

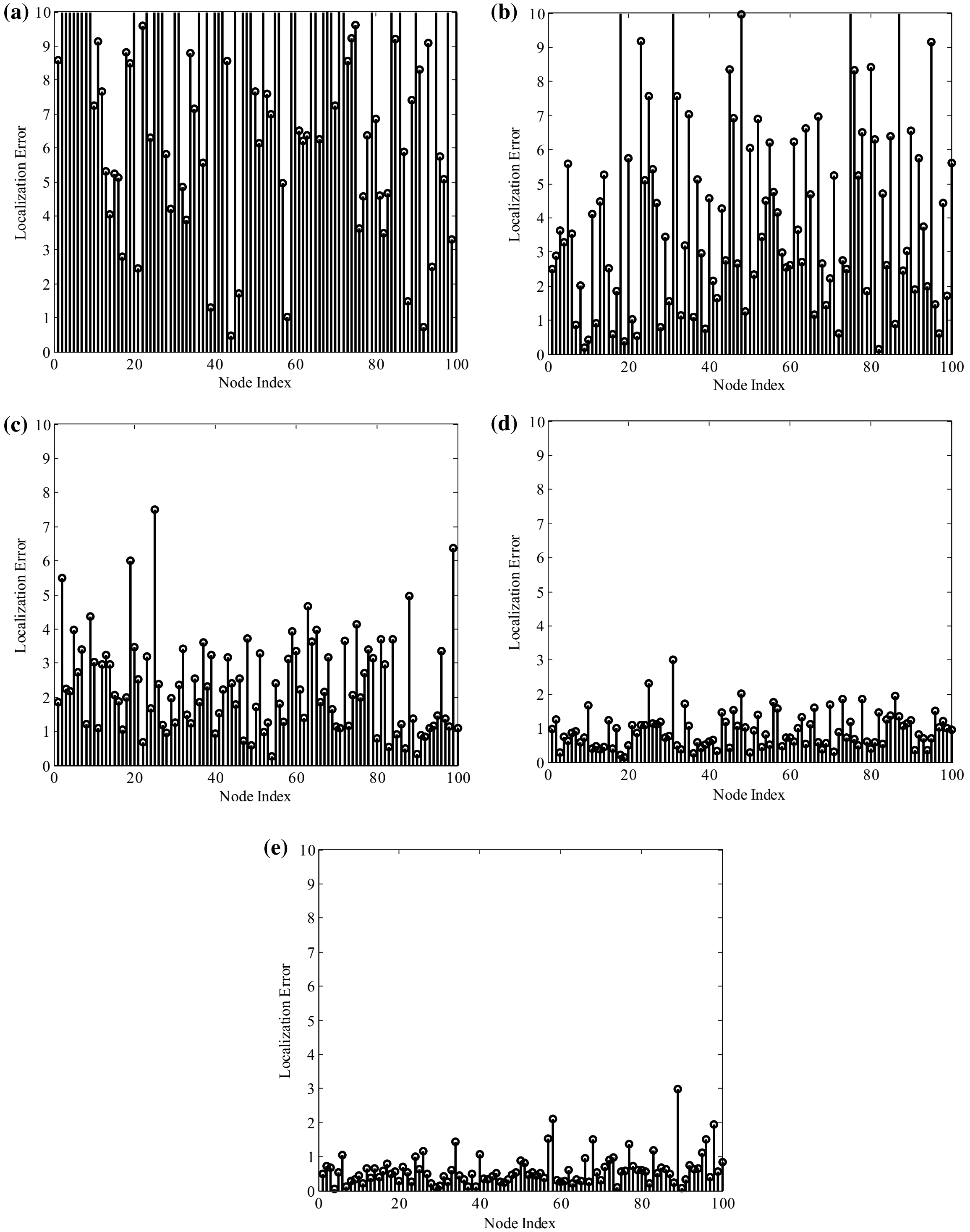

Fig. 3 Localization error of each individual unknown node using various localization techniques: **a** DV-Hop, **b** RSSI, **c** Gauss–Newton, **d** DE algorithm, **e** SA-MCDE algorithm

Table 2 Localization accuracy comparisons for different node densities

Localization technique	Low density		Medium density		High density	
	ALE	RMSE	ALE	RMSE	ALE	RMSE
DV-Hop	12.6727	16.209	11.2641	13.9582	10.355	12.3192
RSSI	4.6046	5.8046	4.2387	5.0683	4.0296	4.8764
Gauss–Newton	2.8498	3.8296	2.6013	2.9224	2.3444	2.7145
DE algorithm	1.6853	1.8536	1.3893	1.5079	1.1373	1.2955
SA-MCDE algorithm	1.0329	1.1313	0.9166	1.0261	0.8311	0.9611

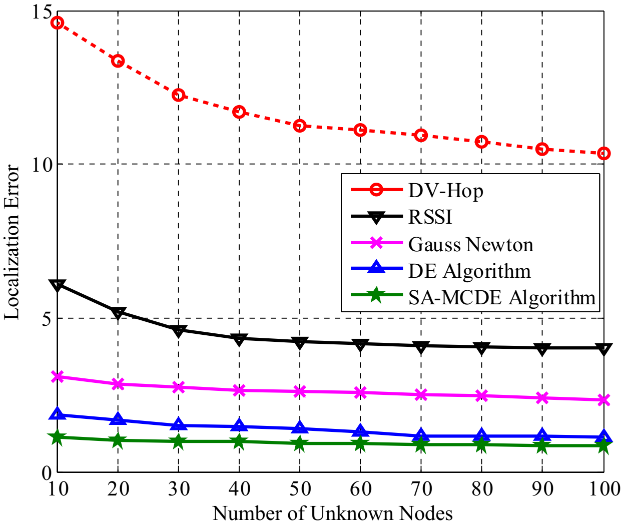

Fig. 4 Localization error for different number of unknown nodes

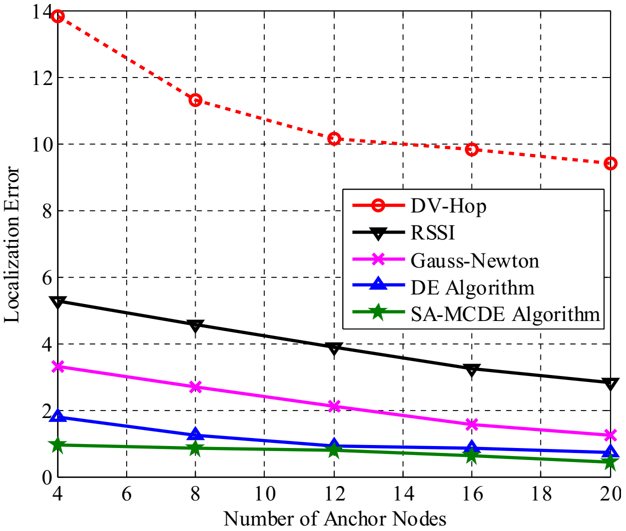

Fig. 5 Localization error for different number of anchor nodes

for different nodes while using DV-Hop and RSSI localization techniques. Because these techniques fail to estimate position of the unknown node which is located far away to all anchor nodes. By contrast, the optimization techniques can optimize the localization problem equally to all nodes irrespective to their locations. Therefore, an uninform error distribution can be observed for optimization techniques based localization as depicted in Fig. 3c–e.

Table 2 describes the accuracy comparisons in terms of both ALE and RMSE among various localization techniques for different load densities. In our simulations, low, medium and high densities are corresponding to deployment of 20, 50, and 100 unknown nodes respectively. As the load density increases, the network connectivity between sensor nodes become high and the number of anchor nodes available within the communication range increases. This leads to decrease localization error as presented in Table 2. This is further observed in Fig. 4, where the ALE of different techniques is presented while increasing number of nodes from 10 to 100. As seen in this figure, the SA-MCDE localization has the minimum localization error compared to all other techniques as the number of nodes increases. Similarly, localization error of various techniques while varying anchor nodes from 4 to 20, keeping number of unknown nodes fixed at 100 is illustrated in Fig. 5. The accuracy of the estimated position is also depends on number of deployed anchors as the connectivity between unknown node and all anchors increases with increased number of anchors. As a result the localization error decreases along with number of anchors as shown in this figure.

At last, the convergence speed in terms of number of iteration or generation to reach minimum least square distance error is measured by using different optimization technique and plotted in Fig. 6. For Gauss–Newton method, the convergence speed is measured with respective to number of iterations and for DE algorithm and SA-MCDE algorithm, the convergence speed is measured with respective to number of generations. The least square distance error is measured from Eq. (1). Though, the Gauss–newton method has high convergence speed, it is frequently trapped by local minima and results high least square error. By contrast, the DE algorithm and SA-MCDE algorithm have an ability to reach global optima and results minimum least square error. Comparing DE algorithm, the convergence speed of SA-MCDE algorithm is further high as depicted in this figure,

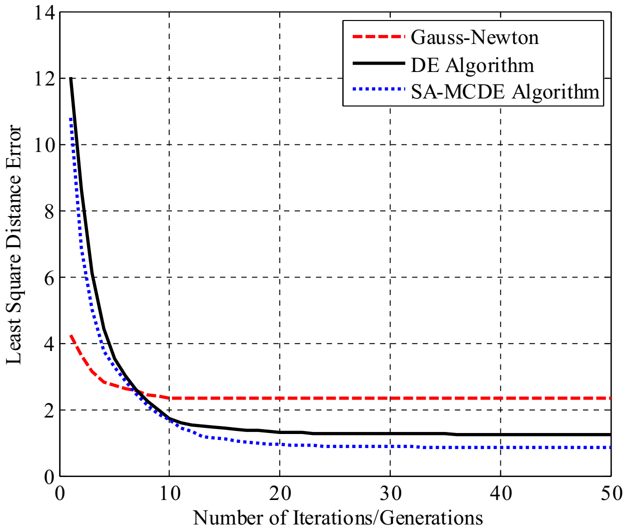

Fig. 6 Convergence speed comparisons among different optimization techniques

because it has a self-controlling mechanism to adjust mutation factor and cross-over probability.

Finally, from the regions simulation study, it is understood that, there is a clear trade-off between range based and range free localization techniques. Thus, localization is defined as LSL problem. During optimizing LSL problem, compared to the Gauss–Newton algorithm the DE algorithm has better localization accuracy in terms of low ALE. The performance and convergence speed of DE algorithm is further improved by including self-adaptive control parameter selection mechanism. Thus, the SA-MCDE algorithm is consistently performing well with high convergence speed under different conditions including at various node densities and for different anchor nodes.

5 Conclusions

Many networking protocol and application require appropriate localization of nodes. The Quality of Service (QoS) of WSN is depends on localization accuracy. In this paper, we investigate SA-MCDE algorithm based localization technique and compared with DE algorithm, Gauss–Newton and classical localization algorithms. Among various localization techniques, the range based techniques have better localization accuracy but they require extra hardware. On the other hand rang free technique are cost effective but they are limited by poor accuracy. So the propose SA-MCDE algorithm localization is found to be better alternative. Through exhaustive simulation study, we found that the SA-MCDE algorithm localization is robust and has better localization accuracy with fast convergence speed.

Generally, MOTs tries to find an optimal solution considering all the controlling parameters. In several problems, these factors are enormous and expressing them in quantity and establishing relationships among these require voluminous calculations. Therefore, developing new MOTs with reduced complexity is always constitutes a promising novel research area. Further, recently, unmanned aerial vehicle (UAV) assisted node localization has become popular as they provide better distance estimates because of clear line of sight between areal anchors and unknown nodes, which may lead to better positioning. Thus, in summary, we can say that localization of WSN continues to be an important research challenge.

References

1. Harl H, Willig A (2005) Protocols and architectures for wireless sensor networks. Wiley, West Sussex
2. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. IEEE Commun Mag 40:102–114
3. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) Wireless sensor networks: a survey. Comput Netw 38:393–422
4. Chizari H, Poston T, Abd Razak S, Abdullah AH, Salleh S (2014) Local coverage measurement algorithm in GPS-free wireless sensor networks. Ad Hoc Netw 23:1–17
5. Aspnes J, Eren T, Goldenberg DK et al (2006) A theory of network localization. IEEE Trans Mob Comput 5:1663–1678
6. Wen F, Liang C (2015) Fine-grained indoor localization using single access point with multiple antennas. IEEE Sens J 15:1538–1544
7. Tarrío P, Bernardos AM, Casar JR (2012) An energy-efficient strategy for accurate distance estimation in wireless sensor networks. Sensors 12:15438–15466
8. Xiong H, Chen Z, Yang B, Ni R (2015) TDoA localization algorithm with compensation of clock offset for wireless sensor networks. China Commun 12:193–201
9. Liu Y, Hu YH, Pan Q (2012) Distributed, robust acoustic source localization in a wireless sensor network. IEEE Trans Signal Process 60:4350–4359
10. Maddumabandara A, Leung H, Liu M (2015) Experimental evaluation of indoor localization using wireless sensor networks. IEEE Sens J 15:5228–5237
11. Xu Y, Zhou J, Zhang P (2014) RSS-based source localization when path-loss model parameters are unknown. IEEE Commun Lett 18:1055–1058
12. Bulusu N, Heidemann J, Estrin D (2000) GPS-less low-cost outdoor localization for very small devices. IEEE Pers Commun 7:28–34
13. Niculescu D, Nath B (2003) DV based positioning in ad hoc networks. Telecommun Syst 22:267–280
14. Liu Y (2011) An adaptive multi-hop distance localization algorithm in WSN. Manuf Autom 33:161–163
15. Cheng BH, Vandenberghe L, Yao K (2009) Distributed algorithm for node localization in wireless ad-hoc networks. ACM Trans Sens Netw 6:8–20
16. Kulkarni R, Venayagamoorthy G, Cheng X, Maggie (2009) Bio-inspired node localization in wireless sensor networks. In: Proceedings of IEEE international conference on systems, man and cybernetics. https://doi.org/10.1109/ICSMC.2009.5346107

17. Arora S, Kaur R (2017) Nature inspired range based wireless sensor node localization algorithms. Int J Interact Multimed Artif Intell 4:7–17
18. Arora S, Singh S (2017) Node localization in wireless sensor networks using butterfly optimization algorithm. Arab J Sci Eng 42:3325–3335
19. Rajakumar R, Amudhavel J, Dhavachelvan P, Vengattaraman T (2017) GWO-LPWSN: grey wolf optimization algorithm for node localization problem in wireless sensor networks. J Comput Netw Commun. https://doi.org/10.1155/2017/7348141
20. Nagireddy V, Parwekar P, Mishra TK (2018) Velocity adaptation based PSO for localization in wireless sensor networks. Evol Intel. https://doi.org/10.1007/s12065-018-0170-4
21. Ni J, Li Z, Qi Y (2018) A gas source localization algorithm based on NLS initial optimization of particle filtering. Evol Intel. https://doi.org/10.1007/s12065-018-0191-z
22. Elbes M, Alzubi S, Kanan T (2019) A survey on particle swarm optimization with emphasis on engineering and network applications. Evol Intel. https://doi.org/10.1007/s12065-019-00210-z
23. Moravec J, Pošík P (2014) A comparative study: the effect of the perturbation vector type in the differential evolution algorithm on the accuracy of robot pose and heading estimation. Evol Intel 6:171–191
24. Harikrishnan R, Jawahar Senthil Kumar V, Sridevi P (2016) A comparative analysis of intelligent algorithms for localization in wireless sensor networks. Wirel Pers Commun 87:1057–1069
25. Annepu V, Rajesh A (2017) An efficient differential evalutionary algorithm based localization in wireless sensor networks. In: Proceedings of international conference on microelectronic devices, circuits and systems. https://doi.org/10.1109/ICMDCS.2017.8211560
26. Storn R, Price KV (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359
27. Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evolut Comput 10:646–657
28. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evolut Comput 13:398–417
29. Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. IEEE Trans Evolut Comput 13:945–958
30. Brest J, Maucec MS (2011) Self-adaptive differential evolution algorithm using population size reduction and three strategies. Soft Comput 15:2157–2174

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.