International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 1 | ISSN : 2456-3307

A Charge System Search based DV Hop Algorithm for Wireless Sensor Networks

Nitin Jain[1], Sushila Madan[2], Sanjay Kumar Malik[3]
[1]Assistant Professor, Department of Computer Science & Engineering, Hindu College of Engineering, Sonepat, Haryana, India
[2]Associate Professor, Department of Computer Science, LSR College for Women, University of Delhi, Delhi, India
[3]Associate Professor, Department of Computer Science & Engineering, SRM University, Delhi-NCR Campus, Sonepat, Haryana, India

ABSTRACT

In Wireless Sensor Networks (WSNs), Localization is one of important issue that can affect the performance of networks. To design an accurate localization algorithm is a challenging task among researchers. It is noticed that DV Hop algorithm gets more attention among all localization algorithms due to its simple and efficient design. But in DV Hop algorithm, the accuracy of positioning is affected through average distance per hop. Hence in this work, an effort is made to improve indoor positioning accuracy and also to reduce positioning error rate in WSNs. So, in this work, charge system search based DV Hop algorithm is proposed for decreasing the error rate due to average distance per hop of anchor nodes. The experimental results reveal that proposed algorithm reduces the positioning error in network effectively.

Keywords : *DV Hop Algorithm, WSNs, GPS, CP, CSS Algorithm*

I. INTRODUCTION

In last few years, a lot of attention and research work have been carried out for successful deployment of sensors in distribute management, information processing, and multi-hop communication [1]. Wireless sensor networks are integrated with advanced technology like sensor technology, embedded computing, communications, and distributed information processing. Large amount of data is gathered through deployed sensors and these sensors are also monitored environmental parameters information collaboratively [2]. Hence, energy consumption and Therefore, collaborative information processing can be considered as key issues in WSNs. But, the collaborative data processing takes less attention among research community. The WSNs are designed using large number of sensor nodes, that can deployed in a sensor field and each node responsible for multi-hop communication [3]. The primary responsibility of these sensor nodes is to collect the data from the sensor field and process the collected data weather belongs to mechanical, thermal, biological, chemical, and optical readings [4]. However, sensors have limited source of power in terms of battery and the power can be consumed due to collection and processing of data. Hence, balanced energy consumption is one of main issue for energy-efficient management [5]. In recent times, lot of research is ongoing for monitoring of residual energy in the field of WSNs. Large numbers of cluster-based routing protocols have been designed for address the above mentioned issue of WSNs. Further, it is observed these nodes are randomly deployed in the sensor field. In most of WSNs applications, it is important that the location of nodes should be known in advance for obtaining sensing data such as target tracking [5], medical care [6] and smart house application [7]. In such application, each node having information about the location of other nodes, without knowing the location information, the collected data have not significant contribution. One of simple solution for addressing aforementioned problem is to localize the nodes using the Global Positioning System (GPS). But, it is not possible in case of sensor nodes, because GPS leads to increase nodes' size, power consumption and cost. Further, it is noticed that GPS devices are not applicable in indoors applications. As a result, few of nodes associated with GPS, called anchors and rest of nodes are called unknown nodes; these are localized through localization algorithms. Hence, node localization is precondition and foundation for the application of WSNs [8].

Presently, location algorithms are divided into two categories i.e. range-based localization algorithm and range-free localization algorithm [9]. The range-based localization algorithms require high power consumption and cost. But, range-free localization algorithms having low cost and power consumption. Additionally, these algorithms do not require any additional hardware support. Thus, these algorithms are widely adopted in WSNs as localization algorithms [10-12]. Among these, DV-Hop algorithm is one of range-free localization algorithm that can be widely adopted as node localization algorithm in WSNs, but, this algorithm suffers with low accuracy. From literature, it is observed that the localization accuracy of DV-Hop algorithm is improved through other meta-heuristic algorithm such as particle swarm optimization algorithm [13], genetic algorithm [14-15], artificial bee colony algorithm [16] and shuffled frog leaping algorithm [17-18] and so on. Hence, the aim of this research work is to explore the capability of charged system search (CSS) algorithm for improving the localization accuracy of DV-Hop algorithm. Kaveh et al., have developed CSS algorithm for solving mechanical design problems in 2010 [19]. This algorithm comprises of three basic principles of physics such as Coulomb law, Gauss law and Newton second law of motion. Every meta-heuristic algorithm consist of two properties i.e. exploration and exploitation. The exploration refers for finding promising solution search space, whereas, exploitation refers to find optimal solution from solution search space. In CSS, Coulomb and Gauss laws are responsible for exploration while Newton second law of motion responsible for exploitation. In this work, CSS algorithm is adopted in second phase of DV-Hop algorithm to optimize hop size error. The advantages of CSS is simple and easy implementation, more accurate results, computational efficiency and less number of user defined parameters.

Rest of paper is organized as fellows. Section 2 describes the DV-Hop algorithm. CSS algorithm is described in section 3. Section 4 describes the proposed improvement in DV-Hop algorithm. In section 5, experimental results are illustrated. Section 6 presents the conclusion of the entire work.

II. DV-HOP ALGORITHM

This section describes DV-Hop algorithm in brief. The distance vector based positioning algorithm for mobile ad-hoc network is proposed by Niculescu and Nath in 2003 [20]. The implementation of algorithm consists of three steps. The very first step of DV-Hop algorithm consists of broadcasting of beacon packets through anchor nodes in network. This packet contains location information with hop count value which is initialized to zero. When a node received a beacon packet, a table is created for every anchor nodes. This table consists of coordinates of anchor nodes and hop size i.e. (x_i , y_i , hop_i) where x_i , y_i represents coordinate of anchor nodes and hop_i represents minimum number of nodes from anchor node. If a received packet having less hop count value to a anchor node, hop count value of the table is updated for received packet, and further, packet is forwarded in the network with increased hop count value by 1; otherwise ignored. Through this mechanism, all nodes in the network maintain the minimum hop count value for every anchor node. After obtaining the minimum hop count for other anchor nodes, the second step is to determine the average size of which is computed through equation 1.

$$\text{hopSize}_i = \frac{\sum_{i \neq j}^{N} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{i \neq j}^{N} h_{ij}} \quad (1)$$

Where, (x_i, y_i) and (x_j, y_j) presents the coordinates of anchor nodes i and j, and h_{ij} is number of hops between anchor nodes i and j.

After computing average hop-size, in next step is to broad cast its hop size through each anchor node in network using controlled flooding. When an unknown node obtains hop-size information message, then, it stored only first received message and after that transmits this message to its neighboring nodes. Using this, most of nodes obtain minimum hop-size from anchor nodes and also compute the distance between itself and anchor nodes using equation 2.

$$d_{ij} = \text{hopSize}_{pi} + h_{ij} \quad (2)$$

Where, hopSize_{pi} denotes the average hop-size of unknown node p obtains from the nearest anchor node_i, and hop_{pk} is the minimum number of hops between the unknown node p and the anchor node k.

After computing the distance between unknown nodes and anchor nodes, the next step is to compute the estimated position of unknown nodes using polygon method. Consider, the position (coordinates) of unknown node p is (x, y), position of i^{th} anchor node is (x_i, y_i), and distance between anchor node_i and unknown node p is d_i. Assuming, an anchor node estimate the location of unknown node p, then, the location is estimated through equation 3.

$$\left.\begin{array}{l} (x - x_1)^2 + (y - y_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = d_2^2 \\ \cdots\cdots\cdots\cdots\cdots \\ (x - x_n)^2 + (y - y_n)^2 = d_n^2 \end{array}\right\} \quad (3)$$

Hence, to make equations linear, subtracting the last equation from first (n–1) equations, then obtained (n – 1) equations as given in equation 4.

$$\left.\begin{array}{l} x_1^2 + y_1^2 - x_n^2 - y_n^2 + 2(x_1 - x_n)x + 2(y_1 - y_n)y = d_1^2 - d_n^2 \\ x_2^2 + y_2^2 - x_n^2 - y_n^2 + 2(x_2 - x_n)x + 2(y_2 - y_n)y = d_2^2 - d_n^2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_{n-1}^2 + y_{n-1}^2 - x_n^2 - y_n^2 + 2(x_{n-1} - x_n)x + 2(y_{n-1} - y_n)y = d_{n-1}^2 - d_n^2 \end{array}\right\} \quad (4)$$

Further, the equation 4 can be rewrite in the form of AX = B, where A, B, and X are described using equations 5-7.

$$A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ 2(x_2 - x_n) & 2(y_2 - y_n) \\ \vdots & \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix} \quad (5)$$

$$B = \begin{bmatrix} x_1^2 + y_1^2 - x_n^2 - y_n^2 + d_1^2 - d_n^2 \\ x_2^2 + y_2^2 - x_n^2 - y_n^2 + d_2^2 - d_n^2 \\ \cdots\cdots\cdots\cdots \\ x_{n-1}^2 + y_{n-1}^2 - x_n^2 - y_n^2 + d_{n-1}^2 - d_n^2 \end{bmatrix} \quad (6)$$

$$X = \begin{bmatrix} x \\ y \end{bmatrix} \quad (7)$$

The location of p^{th} node is computed using equation 8.

$$X = (A'A)^{-1}AB \quad (8)$$

III. CHARGE SYSTEM SEARCH (CSS) METHOD

Kaveh et al., have developed CSS algorithm based on the movement of charged particles in solution search space [19]. The position of i^{th} charged particle (CPs) is represented as $X_i = (X_{i,1}, X_{i,2}, X_{i,3}, \ldots\ldots\ldots\ldots X_{i,D})$.

Whereas, velocity of i^{th} charged particle is given as $V_i = (V_{i,1}, V_{i,2}, V_{i,3}, \ldots\ldots . V_{i,D})$. It is assumed that position and velocities of CPs should lie in the range of $[X_{min}, X_{max}]^D$ and $[V_{min}, V_{max}]^D$ respectively. Initially, the velocities of all CPs are set to zero. From literature, it is noticed that application of CSS algorithm is explored in applied in various research filed and this algorithm provides efficient and effective results for solving many optimization problems [21-24]. The main steps of the CSS algorithm are as follows.

Steps of Charge System Search Optimization Method	
Step 1:	Randomly initialize charge particle in D-dimensional search space using equation 1. $$X_{i,j} = X_{min,i} + r_i * (X_{i,max} - X_{i,min}) \quad (1)$$ $X_{i,j}$ represents the initial value of the i^{th} variable for the j^{th} CP; $X_{min,i}$ and $X_{i,max}$ are the minimum and the maximum values for the i^{th} variable; r_i is a random number in the interval [0,1] and the initial velocities of charged particles are zero i.e. $V_k = 0$, $k = 1, 2, \ldots K$
Step 2:	While (The stopping criterion is not met)
Step 3:	Evaluate fitness of each charged particle, compare the values of fitness function and sort them into increasing order.
Step 4:	Store the positions of initial charged particles (C_k) into a variable, called CM.
Step 5:	For n = 1 to number of charged particles
Step 6:	Compute the value of moving probability for each charged particle using the following equation. $$p_{ik} = \begin{cases} 1 & \dfrac{fit(i) - fit(best)}{fit(j) - fit(i)} \\ 0 & otherwise \end{cases} \quad (9)$$ $$\text{If, } \frac{fit(i)-fit(best)}{fit(j)-fit(i)} > rand \vee fit(j) > fit(i),$$ fit (i) represents the fitness of i^{th} instance of dataset while fit (best) represents the best fitness value and fit (j) represents fitness value of j^{th} data instance of the dataset and rand is a random number in between [0, 1].
Step 7:	Determine the value of actual electric force (F) using the equation 3. $$F_j = q_j \sum_{i,i\neq j} \left(\frac{q_i}{a^3} * i_1 + \frac{q_i}{r_{ij}^2} * i_2\right) * p_{ij} * (X_i - X_j), \begin{cases} j = 1, 2, 3, \ldots K \\ i_1 = 1, i_2 = 0 \leftrightarrow r_{ij} < a \\ i_1 = 0, i_2 = 1 \leftrightarrow r_{ij} \geq a \end{cases} \quad (10)$$ q_i and qj represents the fitness of i^{th} and j^{th} CP, $r_{i,j}$ represents the separation distance between i^{th} and j^{th} CPs, i_1 and i_2 are the two variables whose values are either 0 or 1, 'a' represents the radius of CPs and it is assumed that each CPs has uniform volume charge density but changes in every iteration.
Step 8:	End for
Step 9:	For d = 1 to number of dimensions of charged particles
Step10:	Update the positions and velocities of charged particles using equation (4) and (5) respectively. $$X_{j,new} = rand_1 * Z_a * \frac{F_j}{m_j} * \Delta t^2 + rand_2 * Z_v * V_{j\,old} * \Delta t + X_{j\,old} \quad (11)$$ $$V_{j\,new} = \frac{X_{j\,new} - X_{j\,old}}{\Delta t} \quad (12)$$ $rand_1$ and $rand_2$ are the two random functions whose values lie in between [0, 1], Z_a and Z_v are the control parameters which control the influence of actual electric force and previous velocities, m_j is the mass of j^{th} CPs which is equal to the q_k and Δt represents the time step which is set to 1.
Step 11:	If charged particles cross its boundary limit then correct its position using HS based method.

Step 12: End for

Step 13: Determine the fitness function of new generated charged particles and compare it to the charged particles stored into CM.

Step 14: Exclude the worst charged particles from CM.

Step 15: End while

Step 16: Obtain the desired results

IV. PROPOSED ALGORITHM

This section describes CSS based DV-Hop algorithm for optimizing localization error in second step of original DV-Hop algorithm. In DV-Hop algorithm, an error can occur when compute the estimated distance between unknown nodes and anchor nodes. So, to optimize this error is one of important issue for improving the performance of the localization algorithm. Large numbers of evolutionary algorithm are reported in literature for solving optimization problems. Hence, in this work, CSS based optimization algorithm is adopted to reduce the error rate of DV-Hop algorithm. In literature, it is observed that many researchers have applied evolutionary algorithm for reducing the error rate of DV-Hop algorithm.

4.1 Hop Size Improvement

In DV-Hop algorithm, average hop size is calculated using equation 1and possible contains some error extent. Let us consider that the average hop size for each anchor node is equal to $hopSize_i$ The actual distance between two anchor nodes i and j is denoted using d_{ij} which is computed using equation 2. The error between d_{ij} and the estimated distance which is computed using $hopSize_i$ is illustrated using h_{ij} that denotes the number of hops between two anchor nodes i and j. So, in our proposed algorithm, the CSS algorithm is adopted to optimize the results of first step of DV-Hop, so, the CSS algorithm is applied on the second step of DV-Hop algorithm to reduce the error of determined hop sizes. The fitness function is described in terms of e_{ij} which is correspond to error between real and estimated distance between anchor nodes i and j. The minimization of fitness function e_{ij} is corresponds to better estimation of hop size value for each anchor node. The fitness function can be expressed as given below.

$$f(hopsize_i) = \frac{1}{N-1}\sum_{i \neq j}^{N} \frac{d_{ij}}{h_{ij}} e_{ij}^2 \qquad (13)$$

It also observed that anchor nodes have not equal effects on the error. So, in this work, a weight matrix (W) is computed for each anchor nodes from unknown nodes. This weight matrix is determined using equation 14.

$$W = \begin{bmatrix} w_{p,1} & 0 & \cdots & 0 \\ 0 & w_{p,2} & \cdots & 0 \\ \cdots & \cdots & & \cdots \\ 0 & 0 & & w_{p,n-1} \end{bmatrix} \qquad (14)$$

Where, $w_{p,1}$ is the weight of the unknown node p for i^{th} anchor node and can be computed using equation 15.

$$w_{p,1} = \frac{d_{ij}}{h_{ij}} \qquad (15)$$

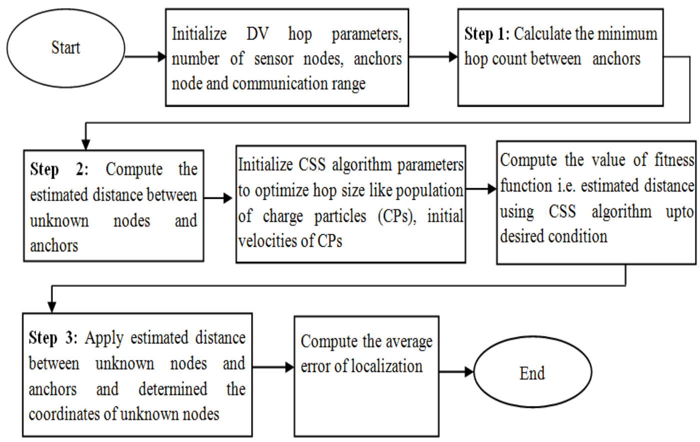

Figure 1. Flow chart of the proposed CSS based DV-Hop algorithm

After determination the value of fitness function, the CSS algorithm is presented. In CSS algorithm, each CPs is described using hop size and each CP contains only one single parameter. Initially, population of CPs are randomly scattered around the computed hop size in the hope to find better optimal solution. Further, the algorithmic steps of CSS algorithm are used to optimize the average ho size and the best CP position can be considered as a new hop size. The flow chart of the proposed CSS based DV-Hop algorithm is illustrated in Figure 1.

V. EXPERIMENTAL RESULTS

5.1. Simulation environment

This subsection describes the simulation environment of proposed CSS based DV-Hop algorithm. The proposed algorithm is implemented in MATLAB environment on window based operating system using 2.20 GHz Intel Core i7 CPU and 8 GB RAM. A 100x 100 sensor field is considered to deploy 100 sensor nodes randomly. The squares nodes represent the anchor nodes whereas circle nodes represent anchor nodes. The communication range of each node is set to 60. The maximum number of iteration is set to 50. The results are taken as the average of twenty five independent runs. The performance of the algorithm is tested on average localization error which is computed using equation 16.

$$\text{Average Error} = \frac{\sum_{i=1}^{M}\sqrt{(x_{estmitaed}-x_i)^2+(y_{estimated}-y_i)^2}}{M} \quad (16)$$

Where, M is the number of unknown nodes, R is communication range, $(x_{estimated}, y_{estimated})$ is the estimated coordinate of unknown node i and (x_i, y_i) is the actual coordinate of unknown node.

Further, the experimental results of the proposed algorithm are compared with DV-Hop algorithm, Ren & Zhao algorithm [15] and Peng & Li algorithm [18].

5.2 Effects of number of anchor nodes on positioning accuracy

Fig. 2 depicts the results of proposed algorithm and other algorithms being compared using average localization error. It is observed that the average error rate is decreasing as the number of anchor nodes increased. Further, it is also noticed that proposed algorithm obtains low error rate among all other algorithms being compared. It is noted that due to increment in number of anchor nodes, hop count between anchor nodes and unknown nodes become decreased and in turn resulted into reduce accumulated error of estimated distance between anchor nodes and unknown nodes.

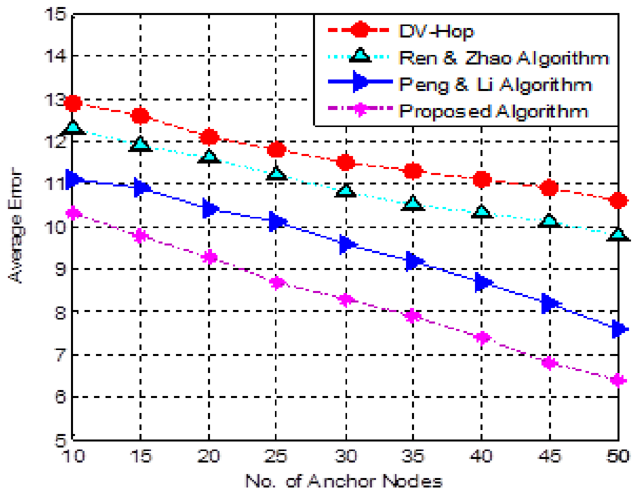

Figure 2. Average localization error of unknown nodes at different number of anchors

5.3 Effects of communication range on positioning accuracy

Figure 3 illustrates the results of average localization error of proposed algorithm, DV-Hop algorithm, Ren & Zhao algorithm and Peng & Li algorithm using communication range. The communication range of each node varies in the range 30 to 60. It is noted that number of sensor nodes and the number of anchor nodes are equal to 100 and 10 respectively. It is observed that error is increased when communication range increases for all algorithms. It is due increment in the hop size. But, it reveals that the proposed algorithm provides less average localization error among all algorithms.

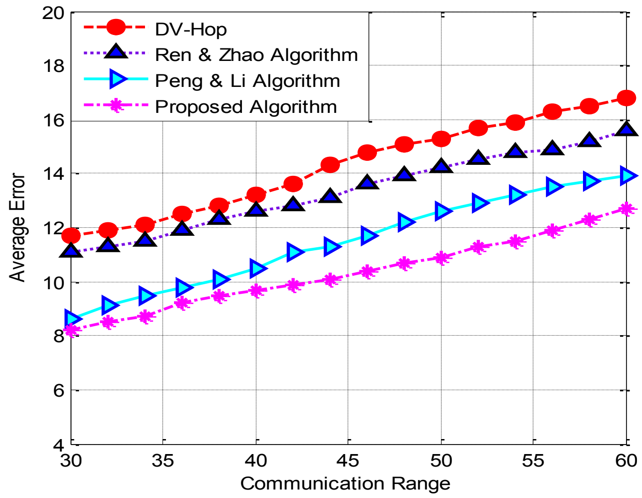

Figure 3. Average localization error of unknown nodes with the change of communication range

5.4 Effects of number of sensor nodes on positioning accuracy

To investigate the performance of the proposed algorithm with increased number of sensor nodes, the numbers of sensor nodes in network are changed from 100 to 300. Still, the communication range is set 60 and the number of anchors is set to 10% of the total nodes. The density of network becomes dense due to increment in number of nodes. Fig. 4 shows the results average localization error of all algorithms using increase number of sensor nodes. It is observed that as the density of network is increased, the number of hops decreased and in turn resulted into more accurate average ho size. Hence, the average localization error can decrease. It is also observed that the proposed algorithm gives better results in comparison to other algorithms being compared.

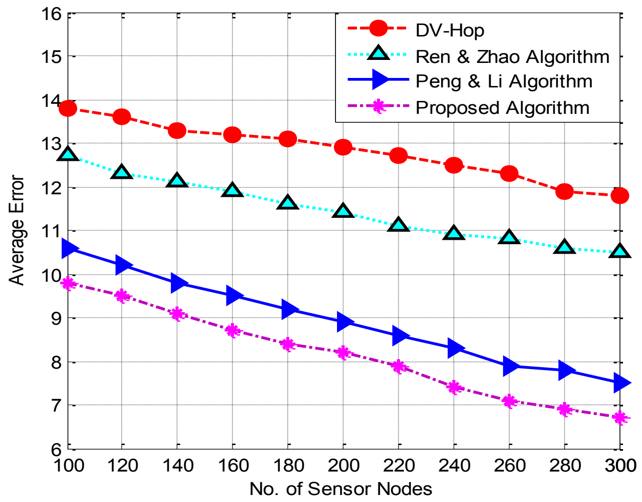

Figure 4. Average localization error of unknown nodes with the change of total nodes

5.6 Computation time

In this work, computational time is also considered as one of performance measure parameter. The computational time of all algorithms is computed in each run. Table 1 describes the average computational time over twenty five independent runs of each algorithm. The number of anchors is set to 10 % of the total numbers of nodes. It is observed that as the numbers of sensor nodes increase, the computational times of all algorithm increases. It is further noted that the computational time of DV-Hop algorithm is lowest among all algorithms.

Table 1. Average computational times of proposed algorithm and other algorithms being compared

Algorithms	Number of Sensor Nodes			
	100	150	200	300
DV-Hop	0.05786	0.09235	0.1256	0.3562
Ren & Zhao algorithm	4.8752	7.8635	9.5673	11.9242
Peng & Li algorithm	8.7312	15.9543	20.0374	24.1457
Proposed	5.0492	7.3124	11.5438	14.6432

VI. CONCLUSION

In wireless sensor networks, locations of sensor nodes have significant impact on its performance in many applications areas. So, in this work, CSS based DV-Hop algorithm is proposed to reduce the localization error for improving the performance of algorithm. The CSS algorithm is applied in the second phase of DV-Hop to reduce average localization error. Further, a weight matrix is also considered for each anchor node because each anchor node imposes different impact on error. The effectiveness of the proposed algorithm is evaluated using three different scenarios i.e. number of anchor nodes, communication range and number of sensor.

From the experimental results, it is stated that the proposed algorithm obtains minimum average localization error in comparison to all other algorithms using all scenarios. Further, computational time is also taken as performance parameter. It is observed that proposed algorithm take more computation time in comparison to DV-Hop algorithm. Finally, it is concluded that effective and efficient algorithm and successfully overcome the shortcoming of DV-Hop algorithm.

VII. REFERENCES

[1]. Tomic, S., & Mezei, I(2016)Improvements of DV-Hop localization algorithm for wireless sensor networksTelecommunication Systems, 61(1), 93–106.

[2]. Yick, J., Mukherjee, B., & Ghosal, D(2008).Wireless sensor network surveyComputer Networks, 52(12), 2292–2330.

[3]. Mesmoudi, A., Feham, M., & Labraoui, N(2013)Wireless sensor networks localization algorithms: A comprehensive survey.International Journal of Computer Networks and Communications (IJCNC), 5(6), 45–64.

[4]. Yang, X., Zhang, W., & Song, Q(2015)An improved DV-Hop algorithm based on shuffled frog leaping algorithmInternational Journal of Online Engineering, 11.

[5]. Lee, S.-M., Cha, H., & Ha, R(2007)Energy-aware location error handling for object tracking applications in wireless sensor networksComputer Communications, 30(7), 1443–1450.

[6]. Pensas, H., Raula, H.,&Vanhala, J(2009)Energy efficient sensor network with service discovery for smart home environmentsIn Third international conference on sensor technologies and applications,2009 (SENSORCOMM'09)IEEE.

[7]. Chen,Y., et al(2010).Asmart gateway for health care system using wireless sensor networkIn 2010 fourth international conference on sensor technologies and applications (SENSORCOMM)IEEE.

[8]. Niclescu, D., NLAmericaCommunication Paradigms for Sensor Network– IEEE Communications Magazine, Vol43, 2005, No 3, pp116-122.

[9]. Sun, L., JLi, YChenWireless Sensor NetworksTsinghua University Press, Beijing, 2005, pp45-50.

[10]. Arivubrakan P., VRSDhulipalaEnergy Consumption Heuristics in Wireless Sensor Networks– In: Procof IEEE International Conference on Computing, Communication and Applications, Din Digul, Tamilnadu, 2012, pp1-3.

[11]. Ji, B., LWang, QYangNew Version of AES-ECC Encryption System Based on FPGA in WSNs– Journal of Software Engineering, Vol9, 2015, No 1, pp87-95.

[12]. Arivubrakan, P., VRSDhulipalaSentry Based Intruder Detection Technique for Wireless Sensor Networks– Journal of Artificial Intelligence, Vol6, 2013, No 2, pp175-180.

[13]. Huang, Y., C.-Z., Zang, H.-BYuLocalization Method Based on Modified Particle Swarm Optimization for Wireless Sensor Networks– Control and Decision, Vol27, 2012, No 1, pp156-160.

[14]. Zhang, W., QSongAn Improved DV-Hop Algorithm Based on Genetic Algorithm– Journal of Chongqing University, Vol38, 2015, No 3, pp162-169.
[15]. Ren, W., & Zhao, C(2013)A localization algorithm based on SFLA and PSO for wireless sensor networkInformation Technology Journal, 12(3), 502.
[16]. Li, M.-D., WXiong, LGuoImprovement of DV-Hop Localization Based on Artificial Bee COLONY Algorithm– Computer Science, Vol40, 2013, No 1, pp33-36.
[17]. Ge, Y., X.-PWang, JLiangImprovement of DV-Hop Localization Based on Shuffled Frog Leaping Algorithm, Journal of Computer Applications, Vol31, 2011, No 4, pp922-924, 1002.
[18]. Peng, Bo, & Li, Lei(2015)An improved localization algorithm based on genetic algorithm in wireless sensor networksCognitive Neurodynamics, 9(2), 249–256.
[19]. Kaveh, A., and Talatahari, S(2010 a)A novel heuristic optimization method: charged system searchActa Mechanica, vol213, no3-4, pp267-289.
[20]. Niculescu, D., & Nath, B(2003)DV based positioning in ad hoc networksTelecommunication Systems, 22(1–4), 267–280.
[21]. Kaveh, A., & Laknejadi, K(2011)A novel hybrid charge system search and particle swarm optimization method for multi-objective optimizationExpert Systems with Applications, 38(12), 15475-15488.
[22]. Kumar, Y., & Sahoo, G(2014)A charged system search approach for data clusteringProgress in Artificial Intelligence, 2(2-3), 153-166.
[23]. Kumar, Y., & Sahoo, G(2014)A chaotic charged system search approach for data clusteringInformatica, 38(3), pp249-261.
[24]. Kumar, Y., & Sahoo, G(2015)Hybridization of magnetic charge system search and particle swarm optimization for efficient data clustering using neighborhood search strategySoft Computing, 19(12), 3621-3645.