

# Nature Inspired Algorithm-Based Improved Variants of DV-Hop Algorithm for Randomly Deployed 2D and 3D Wireless Sensor Networks

Amanpreet Kaur<sup>1</sup>  · Padam Kumar<sup>1</sup> · Govind P. Gupta<sup>2</sup>

Published online: 12 April 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

**Abstract** Localization is a significant challenge in the area of wireless sensor networks (*WSNs*). Distance Vector Hop (*DV-Hop*) algorithm is most preferable algorithm due to its low cost, distributed nature, and its feasibility for all kinds of sensor networks, but it suffers from high localization error. In order to reduce the problem of high localization error for 2-dimensional and 3-dimensional *WSNs*, two Nature Inspired Algorithm based improved variants have been proposed. The first one uses Grey-Wolf optimization (*GWO-DV-Hop*) to identify a better estimate of average distance per hop and second one, a weighted Grey-Wolf optimization (*Weighted GWO-DV-Hop*), finds average distance per hop as computed by each beacon node using grey wolf algorithm and then, a weighted approach is applied by each node to get weighted average distance per hop (weights based on distance from each beacon) so as to consider impact of all types of beacons. The results prove the superiority of proposed algorithms over traditional *DV-Hop* in terms of localization error.

**Keywords** Wireless sensor networks · Localization · Grey wolf algorithm · Average distance per hop · Localization error

## 1 Introduction

Wireless sensor network (*WSN*) is an undeniably essential innovation that attracts considerable research interest [1]. A typical *WSN* comprises large number of nodes with implicit sensors which are engaged in sensing the events that occur in the surrounding environment [2]. Localization is crucial challenge in *WSNs*, as location information of an event is also required along with the sensing information in several types of applications,

---

✉ Amanpreet Kaur  
amanpreet.kaur1410@gmail.com

<sup>1</sup> Department of Computer Science and Engineering, Jaypee Institute of Information Technology, Noida, India

<sup>2</sup> Department of Information Technology, National Institute of Technology, Raipur, India

such as wild life surveillance, fire surveillance, water quality monitoring, intrusion detection, health monitoring and precision agriculture etc. In these applications, the estimation/event information is not much use unless the location where the event has taken place is available [3]. Localization algorithms determine the position of nodes by exploiting knowledge of absolute positions of few nodes called beacons. GPS can be used as solution for localization, but it is not a feasible solution because of its size, energy consumption and high cost.

The localization algorithms are of two kinds: range-based or range-free [4]. Range-based algorithm [5–9] gives high precision due to use of special equipment which increases its cost, although its estimates are affected by noise and multi-path fading. On the other hand, range-free algorithms [10–14] require connectivity information and no specialized hardware, thus they are cost effective. Most of the researchers have paid higher attention towards range-free rather than range-based technique due to its low cost. In previous studies, localization algorithms mainly focus on 2D space, but generally, nodes are deployed in 3D space for many applications. Thus, localization technique should be developed for 3D space to make it more realistic. Among range-free algorithms, *DV-Hop* algorithm is extremely popular because of its simplicity, cost-effectiveness, and robustness, but still it has low accuracy. Therefore researchers are still working on improving this algorithm.

In this paper, we put forward the application of Grey-Wolf-based optimization technique with *DV-Hop* algorithm in both 2D and 3D WSN. We have proposed two algorithms that use Grey-Wolf-based optimization technique. Simulation results prove that Grey-Wolf-based optimization technique if merged with DV-Hop, results in high localization accuracy.

The paper is structured as: Sect. 2 discusses the localization problem. Section 3 gives the introduction of *DV-Hop* algorithm and its error analysis. Section 4 discusses the overview of Grey-Wolf optimization. Section 5 is followed by implementation of proposed algorithms. The simulation is presented in Sect. 6. Then, Sect. 7 discusses concluding remarks and future work.

## 2 Problem Statement

Localization is the process of obtaining absolute coordinates of nodes in a WSN [15]. In this problem, we take a WSN having  $N$  total nodes. There are three varieties of nodes.

- (a) *Beacon nodes*—Those that know their position.
- (b) *Settled nodes*—Those that initially are unaware of their location, but then are able to get this information by applying localization technique.
- (c) *Unsettled nodes*—Those that are unaware of their location.

Assuming there are  $m$  beacon nodes and  $n$  unsettled nodes ( $m$  is at least 10% of  $N$ ), the localization problem is to convert these  $n$  unsettled nodes to settled ones.

## 3 Related Work

Here, we present a brief overview of *DV-Hop* and its various improved versions proposed in the literature.

### 3.1 Traditional DV-Hop Localization Scheme

DV-Hop was first put forwarded by Niculescu et al. [11] in 2003. It is a distributed hop counting technique. The symbol table used for DV-Hop Algorithm applied for 3D space is given in Table 1.

This algorithm works in three phases for 3D space that are described as follows:

*Phase 1:* All beacon nodes pass packets containing their coordinate information  $(x_i, y_i, z_i)$  along with their id  $i$ , hop count( $hop_i$ , initially set to 0) to all other nodes within their communication range. All the nodes maintain their own hop count table containing ids, position information and hop counts of all beacons in the network with the help of these packets. The receiving node first increments the hop count value and then updates it in its table for given beacon  $i$  if it is the first packet received from beacon, otherwise it verifies the hop count value from its table and if the received incremented value is smaller than the value stored in the table, then the node replaces the stored value by received value. This phase allows every node to know minimum hop counts from every beacon in the network.

*Phase 2:* After the first phase, all the beacons reckon average distance in one hop ( $AvgHopSize_i$ ) by utilizing Eq. (1).

$$AvgHopSize_i = \frac{\sum_{j=1, i \neq j}^m \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}}{\sum_{i=1, i \neq j}^m hop_j} \tag{1}$$

where  $i$  is the id of beacon  $i$ ,  $(x_i, y_i, z_i)$ ,  $(x_j, y_j, z_j)$  are coordinates of beacons  $i$  and  $j$ ,  $j$  lies between 1 to  $m$  ( $j \neq i$ ),  $hop_j$  is the hop count of beacon  $j$  from beacon  $i$ .

Then, they send one more packet containing calculated  $AvgHopSize_i$  to all other nodes. A receiving one (assuming having id  $u$ ) computes approximate distances ( $d_j$ ) from each beacon node  $j$  by multiplying  $AvgHopSize_i$  of its nearest beacon  $i$  by hop count ( $hop_j$ ) obtained from its hop count table [Eq. (2)].

$$d_j = AvgHopSize_i \times hop_j \tag{2}$$

*Phase 3:* This is the last phase applied by unsettled nodes to move to settled status. They apply trilateration/multilateration method to get their absolute coordinates. The whole process is explained below:

Let the coordinates of unsettled node  $u$  and beacon  $B_j$  be  $(x_u, y_u, z_u)$  and  $(x_j, y_j, z_j)$   $\{j \in (1, m)\}$ . The following system of equations can be formed:

**Table 1** Symbol table for DV-Hop algorithm

Symbol	Description
$i$	Identity of all nodes (in number format)
$(x_i, y_i, z_i)$	x, y and z coordinates of nodes
$hop_i$	Hop count from beacon $i$
$AvgHopSize_i$	Average hop distance computed by beacon $i$
$N$	Count of nodes in WSN
$n$	Count of unsettled nodes
$m$	Count of beacons

$$\left. \begin{aligned} (x_u - x_1)^2 + (y_u - y_1)^2 + (z_u - z_1)^2 &= d_1^2 \\ (x_u - x_2)^2 + (y_u - y_2)^2 + (z_u - z_2)^2 &= d_2^2 \\ &\vdots \\ &\vdots \\ (x_u - x_m)^2 + (y_u - y_m)^2 + (z_u - z_m)^2 &= d_m^2 \end{aligned} \right\} \tag{3}$$

Equation (4) can be derived from Eq. (3).

$$\left. \begin{aligned} -x_m^2 + y_1^2 - y_m^2 + z_1^2 - z_m^2 - d_1^2 - d_m^2 &= 2 \times x_u \times (x_1 - x_m) + 2 \times y_u \times (y_1 - y_m) + 2 \times z_u \times (z_1 - z_m) \\ x_2^2 - x_m^2 + y_2^2 - y_m^2 + z_2^2 - z_m^2 - d_2^2 - d_m^2 &= 2 \times x_u \times (x_2 - x_m) + 2 \times y_u \times (y_2 - y_m) + 2 \times z_u \times (z_2 - z_m) \\ &\vdots \\ x_{m-1}^2 - x_m^2 + y_{m-1}^2 - y_m^2 + z_{m-1}^2 - z_m^2 - d_{(m-1)}^2 - d_m^2 &= 2 \times x_u \times (x_{m-1} - x_m) + 2 \times y_u \times (y_{m-1} - y_m) + 2 \times z_u \times (z_{m-1} - z_m) \end{aligned} \right\} \tag{4}$$

The matrix form of Eq. (4) is Eq. (5) as given below:

$$AX_u = B \tag{5}$$

where  $A = 2 \times \begin{bmatrix} x_1 - x_m & y_1 - y_m & z_1 - z_m \\ x_2 - x_m & y_2 - y_m & z_2 - z_m \\ \vdots & \vdots & \vdots \\ x_{m-1} - x_m & y_{m-1} - y_m & z_{m-1} - z_m \end{bmatrix}$ ,  $X_u = \begin{bmatrix} x_u \\ y_u \\ z_u \end{bmatrix}$ , and

$$B = \begin{bmatrix} x_1^2 - x_m^2 + y_1^2 - y_m^2 + z_1^2 - z_m^2 - d_1^2 - d_m^2 \\ x_2^2 - x_m^2 + y_2^2 - y_m^2 + z_2^2 - z_m^2 - d_2^2 - d_m^2 \\ \vdots \\ x_{m-1}^2 - x_m^2 + y_{m-1}^2 - y_m^2 + z_{m-1}^2 - z_m^2 - d_{m-1}^2 - d_m^2 \end{bmatrix}$$

Equation (5) can be transformed to Eq. (6) as follows:

$$X_u = (A^T A)^{-1} A^T B \tag{6}$$

Equation (6) is solved to obtain the coordinates of the node by using Least Square (LS) method.

The algorithm is exceptionally prominent algorithm as it is very simple, feasible, scalable and applicable for nodes having less than 3 neighbour nodes. The nodes that are having less than 3 neighbours can also be localized using this algorithm due to its distributed nature. But its accuracy is affected by several factors.

The algorithm has served as motivation for many range free localization algorithms. Extensive research has been done in enhancing its accuracy. In [16], Tomic proposed three distinct kinds of algorithms (iDV-Hop1, iDV-Hop2, and Quad DV-Hop). The first two algorithms add one more step which includes geometry techniques which consider the impact of nearest beacon to increase the accuracy. The third algorithm replaces the third phase by applying non-linear method called quadratic programming solver to increase the accuracy. Two more improved methods were proposed in [17] by Song et al. that has utilized weighted method in place of leration method in third phase. In second algorithm, he used a non-linear method called hyperbolic algorithm in the third phase. The first algorithm reduces the complexity of DV Hop with a small improvement in accuracy, whereas, second approach improves the accuracy to a large extent. In [18], Zhang introduced two algorithms which apply weighted approach in the final phase. Both algorithms choose the weight that is inversely proportional to number of hops. The concept in these

algorithms was that the beacon which is closer to the node should have higher role than the beacon that is far from it. Both these algorithms achieve higher accuracy. In [19], Fang added a compensation coefficient to correct distance between beacons and regular nodes. An improvement was made in [20] by optimizing location of node in third phase of DV-Hop algorithm by applying genetic algorithm which results in higher accuracy. In [21], Gui et al. introduced a small low computational step as fourth phase to improve the accuracy of algorithm. Gui et al. also proposed another algorithm in [21] which replaces third phase with very complex method selecting 3 best beacons for each particular node to determine its location. In [22], Shahzad et al. computed the optimal value of  $\max\_hop$  that is used by node to determine its location. If the beacons are within this  $\max\_hop$  value, then only that beacon is selected by the node to get its location, otherwise it is discarded.

In [23], Wang et al. introduced a weighted algorithm which also uses Genetic algorithm and simplex method in its third phase to solve the problem of high localization error of weighted DV-Hop. Experiments prove that localization accuracy is improved significantly by applying hybrid algorithm of genetic algorithm and simplex method in third phase of DV-Hop algorithm. Kumar et al. [24] also proposed an improved algorithm which uses an improvement term to refine the hop-size of anchor nodes in second phase of DV-Hop and uses unconstrained optimization technique in place of least square method in third phase to attain better localization accuracy.

These entire algorithms have tried to improve localization accuracy to some extent, but still there is scope to improve the accuracy further. Based on this fact, we are proposing an improved variant of DV-Hop by refining the value of average hop distances computed by all beacons which is the root cause of localization error.

#### 4 Error Analysis of DV-Hop Localization Scheme

The main factor which contributes most to localization error is the error in computing average hop distance by each beacon. The beacon  $i$  compute its exact distance from any beacon  $j$  ( $d_j$ ) by applying following Eq. (7).

$$d_j = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (7)$$

If  $AvgHopSize_i$  is average distance per hop for beacon  $i$ , then its approximate distance ( $d1_j$ ) from beacon  $j$  is product of  $hop_j$  and  $AvgHopSize_i$  [Refer Eq. (3)]. The error  $e_i$  is the difference in these two values ( $e_i = d_j - d1_j$ ). The optimized  $AvgHopSize_i$  should reduce the value of  $e_i$  as small as possible. To reduce this error in computing average hop size, grey wolf optimization algorithm is applied.

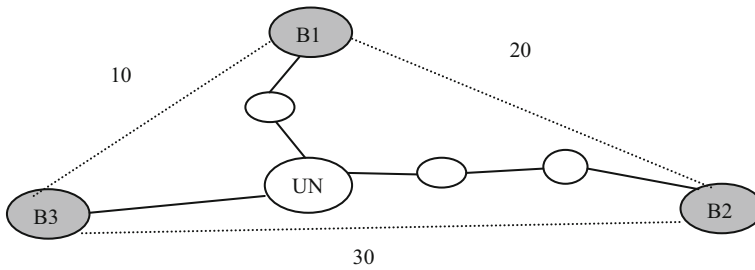
The objective function used in grey wolf optimization is described as follows:

$$\min e_i = \sum_{j=1, j \neq i}^m (d_j - d1_j) \quad (8)$$

$$0 < AvgHopSize_i \leq \max d_j$$

where  $d1_j = AvgHopSize_i \times hop_j$ . The problem is to find optimum value of average hop size of beacon  $i$  by minimizing the above function.

Consider a WSN network as shown in Fig. 1 in which there are three beacons  $B_1$ ,  $B_2$  and  $B_3$  and unsettled node ( $UN$ ) that wants to locate itself. All these beacons know their



**Fig. 1** Example of DV-Hop algorithm

distances from each other after phase 1 of DV-Hop. Let the distances be 10, 20 and 30 respectively.  $B_1$  is 2 hops apart from UN,  $B_2$  is 3 hops apart from UN and  $B_3$  is 1 hop apart from UN. Let the length of each hop be 5 m and  $B_3$  is 10 m apart from UN. In the phase 2, all Beacon nodes compute their AvgHopSizes by applying Eq. (1).

$$AvgHopSize_1 = \frac{10 + 20}{3 + 5} = 3.75$$

$$AvgHopSize_2 = \frac{20 + 30}{5 + 4} = 5.56$$

$$AvgHopSize_3 = \frac{10 + 30}{3 + 4} = 5.71$$

The beacon nodes pass average hop distances to all other nodes. The unsettled node picks the first received average hop size and discards other. The UN node chooses average hop size of  $B_3$  node as it is just 1 hop apart from it. Thus according to DV Hop phase 2,  $B_3$  is 5.71 m apart from UN, although actual distance is 10 m. Thus, localization error is  $10 - 5.71 = 4.29$  m, that is almost 40% of actual distance. Due to this high error, the UN node localizes itself incorrectly. From above analysis, we infer that the localization error of the DV-Hop is influenced by approximate calculation of average hop sizes by beacons. Thus, to reduce localization error, the average hop sizes need to have better approximation.

## 5 Overview of the Grey-Wolf Optimization (GWO) Algorithm

*Grey-Wolf* optimization algorithm was proposed by Mirjalili et al. [25]. It is a new meta-heuristic optimization technique inspired by hunting mechanism of grey wolves. The reason to select this algorithm is that it is very simple, flexible and it tries to search entire search space rather than selecting only local optima solution. In [25], the given algorithm has been compared with other well known optimization algorithms for 28 benchmark functions and it has been proved that grey-wolf is superior to other meta-heuristic optimization algorithms. GWO algorithm depends on the classification of grey wolves. The grey wolves are of 4 kinds. The first category called alphas that are leaders to help in taking choices about chasing, rest place, time to wake and many more. The next category is of betas that are subordinates of alpha. The lowest in this hierarchy are omegas and deltas.

For GWO, we consider that the best solution is alpha ( $\alpha$ ). The next two best solutions are beta ( $\beta$ ) and delta ( $\delta$ ). The rest solutions are omega ( $\omega$ ). During hunting, the wolves encircle the prey which can be depicted using following mathematical model:

$$D = C \cdot X_p(i) - X(i) \quad (9)$$

$$X(i+1) = X_p(i) - A \cdot D \quad (10)$$

where  $i$  is the iteration,  $A$  and  $C$  are coefficient vectors,  $X_p$  is the position of prey and  $X$  is the position vector of grey wolf. The vectors  $A$  and  $C$  are computed as follows:

$$A = 2a \cdot r_1 - a \quad (11)$$

$$C = 2 \cdot r_2 \quad (12)$$

where  $a$  is a vector which decreases linearly from 2 to 0 and  $r_1, r_2$  are random vectors in range from 0 to 1.

In GWO, it is assumed that the alpha, beta and delta knows better about prey. Thus, following formulas are proposed to get the location of prey.

$$D_\alpha = C_1 \cdot X_\alpha - X, \quad D_\beta = C_2 \cdot X_\beta - X, \quad D_\delta = C_3 \cdot X_\delta - X \quad (13)$$

$$X_1 = X_\alpha - A_1 \cdot D_\alpha, \quad X_2 = X_\beta - A_1 \cdot D_\beta, \quad X_3 = X_\delta - A_1 \cdot D_\delta \quad (14)$$

$$X(i+1) = \frac{X_1 + X_2 + X_3}{3} \quad (15)$$

The pseudo code of GWO is depicted in Fig. 2.

## 6 Proposed Algorithms

In this section, we present two algorithms, called Grey wolf optimization based DV-Hop (GWO-DV-Hop), and weighted grey wolf optimization based DV-Hop (Weighted GWO-DV-Hop). They work for both 2D and 3D space, although we describe below for 3D space only.

---

### Algorithm 1 Grey Wolf Optimization Algorithm

---

```

Initialize the population of grey wolf,  $a$ ,  $A$  and  $C$ 
Determine the fitness function of each search agent
Let  $X_\alpha$  = the best search agent,  $X_\beta$  = the second best search agent and  $X_\delta$ 
= the third best search agent
for all  $i$  in  $maxiterations$  do
  for each search agent do
    Update the position of the search agent by applying Equ.10
  end for
  Update  $a$ ,  $A$  and  $C$ 
  Again, determine the fitness function
  Update  $X_\alpha$ ,  $X_\beta$  and  $X_\delta$ 
end for
return centroid of  $X_\alpha$ ,  $X_\beta$  and  $X_\delta$ .

```

---

**Fig. 2** Steps used in the Grey wolf optimization algorithm

## 6.1 GWO-DV-Hop Algorithm

The proposed GWO-DV-Hop localization algorithm works in three phases. The assumptions are: circular radio range, symmetric radio connectivity, no additional hardware, no obstructions, lack of line-of-sight, no multipath and fading.

*Phase 1: Calculating minimum hop counts from each beacon:* Each node gets position and minimum number of hop counts from each beacon by applying broadcasting mechanism. The phase is similar to the phase 1 of the traditional DV-Hop.

*Phase 2: Refinement of Average Hop Distance by the beacon:* In this phase, the beacons calculate their average hop sizes by applying Eq. (1) and then compute their distances from each beacon by applying Eq. (7). After this, the beacons use grey wolf algorithm to refine average hop distances. The fitness function used to achieve optimized average hop distance is mentioned by Eq. (8). This phase helps all the beacons to get their refined average hop sizes. All these average hop sizes are broadcasted to all other nodes in the network who then derive their distances from every beacon by taking the product of refined average hop size of nearest beacon to the given node and minimum hop counts. This distance information is used in the next phase. The steps of using grey wolf algorithm for obtaining the optimum average hop size by every beacon are given in Fig. 3.

*Phase 3: Converting the unsettled node to settled node:* The unsettled nodes locate themselves by applying lateration method [11] such as least square method as described in Phase 3 of Sect. 3.1.

## 6.2 Weighted GWO-DV Hop Algorithm

*Phase 1: Calculating minimum hop counts from each beacon:* Same as in 6.1.

*Phase 2: Refinement of Average Hop Distance by the beacon:* Same as Phase 2 in GWO-DV-Hop.

*Phase 3: Converting the unsettled node to settled node:* The unsettled nodes compute their own average hop distance by considering impact of average hop distances of all the beacons. The beacon that is closer to the node has higher weight than the beacon that is far away from the node. The average hop distance computed by the node  $u$  is described using Eq. (16).

$$AvgHopSize_u = \frac{\sum_{i=1}^m w_i \times AvgHopSize_i}{\sum_{i=1}^m w} \quad (16)$$

where  $w_i = \frac{1}{hop_i}$  is the weight of each beacon  $i$ .

Then, the node  $u$  obtain its distances from every beacon (say with id  $i$ ) by applying Eq. (17).

$$d_i = AvgHopSize_u \times hop_i \quad (17)$$

After obtaining all these distances from every beacon, the unsettled node settles itself by applying lateration method [11] which uses least square method.



**Algorithm 2** Calculation of the optimum average hop size using grey wolf optimization algorithm

---

```

Initialize a random population in the space between 0 and max ( $d_j$ ), value of
 $a$ ,  $A$  and  $C$ .
Find the objective function  $fobj$ ,  $fobj\beta$ ,  $fobj\delta$  (Use Equ. 8.
Set the initial best estimates of  $X\alpha$ ,  $X\beta$  and  $X\delta$ . In this algorithm, the  $X\alpha$ 
is set to average hop size computed by given beacon,  $X\beta$  and  $X\delta$  are set to
average of average hop sizes computed by all beacons.
Initialize  $i = 1$ .
for all  $i$  in  $maxiterations$  or  $fobj \geq error$  ( $i$  represents iteration number)
do
  for each beacon do
    Update its average hop size by using Eq. (15). Eq. (15) uses Equations
    (13) and (14)
  end for
  Update the value of  $a$ ,  $A$  and  $C$ .
  Again determine the objective function  $fobj1$ .
  if  $fobj1 \leq fobj$  then
    then update  $X\alpha$  with new value and  $fobj = fobj1$ 
  else
    if  $fobj1 > fobj$  and  $fobj1 < fobj\beta$  then
      update  $X\beta$  with new value and  $fobj\beta = fobj1$ 
    end if
  else
    if  $fobj1 > fobj$  and  $fobj1 > fobj\beta$  and  $fobj1 < fobj\delta$  then
      update  $X\delta$  with new value and  $fobj\delta = fobj1$ 
    end if
  else
    discard the value of  $fobj1$  and retain previous  $fobj$ 
  end if
  Update  $X\alpha$ ,  $X\beta$  and  $X\delta$  using Equ. 13.
   $i = i + 1$ 
end for
get centroid of  $X\alpha$ ,  $X\beta$  and  $X\delta$  and return the optimal average hop size as
this value.

```

---

**Fig. 3** Phase 2 of proposed algorithm

## 7 Results and Analysis

This section presents a detail performance comparison of the proposed algorithm with the state-of-art DV-Hop algorithm and discusses its performance by varying different network parameters. Next, complexity analysis of the proposed algorithm with the DV-Hop are listed and discussed.

### 7.1 Performance Evaluation and Discussions

To evaluate the performance of the proposed algorithms, we implemented all the algorithms and compared their performance on the basis of Localization Error (in %). The algorithms are implemented in both 2D and 3D environment. The parameters used for the simulation are described in Table 2 for 2D space and in Table 3 for 3D space. The minimum communication radius for 2D is set to 15 m and for 3D is 20 m. It is observed that the value of communication radius depends upon the node density. Smaller the node

**Table 2** Simulation parameters for 2D space

Simulation parameters	Value
WSN area	Vary from $100 \times 100$ to $300 \times 300 \text{ m}^2$
Total nodes ( $N$ )	Vary from 200 to 450
Beacon nodes ( $m$ )	Vary from 20 to 80
Total iterations	50
Communication radius ( $r$ )	Vary from 15 to 35

**Table 3** Simulation parameters for 3D space

Simulation parameters	Value
WSN Area	Vary from $100 \times 100 \times 100$ to $300 \times 300 \times 300 \text{ m}^3$
Total Nodes( $N$ )	Vary from 200 to 450
Beacon Nodes( $m$ )	Vary from 20 to 80
Total iterations	50
Communication radius( $r$ )	Vary from 20 to 40

density, larger is the value of communication radius so that all the nodes are able to communicate with each other. This is the reason that the minimum value of communication radius for 3D is greater than that used in 2D space. The percentage of beacons used should be 10% of total nodes.

The localization accuracy [26–33] is formulated as follows:

$$\text{localization error} = \frac{1}{n \times r} \sum_{i=1}^n \sqrt{(x_a - x_u)^2 + (y_a - y_u)^2 + (z_a - z_u)^2} \quad (18)$$

where  $n$  is the total number of unknown nodes,  $(x_a, y_a, z_a)$  are the actual coordinates and  $(x_u, y_u, z_u)$  are the estimated coordinates of unknown node and  $r$  is the communication radius of sensors.

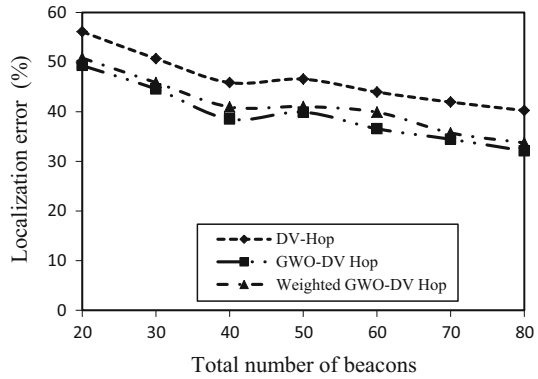
The localization accuracy enhances as the localization error reduces.

It can be seen from Figs. 4, 5, 6, 7, 8, 9, 10, and 11 that grey-wolf based perform better in terms of localization accuracy. Figures 4, 5, 6, and 7, represent results for 2D space, and Figs. 8, 9, 10, and 11 represent results for 3D space.

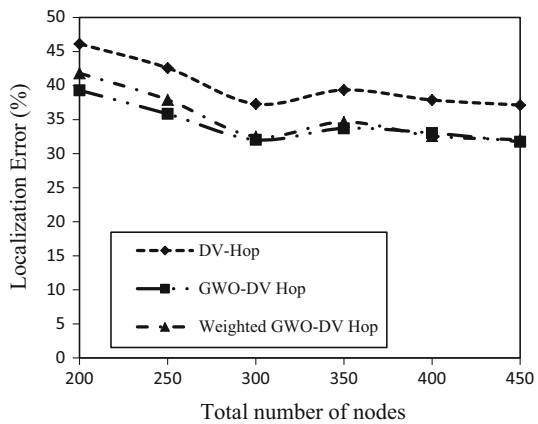
Figure 4 illustrates that the proposed algorithms reduces the localization error when the percentage of beacon nodes is increased from 10 to 40% for WSN area  $100 \times 100$  and total nodes = 200. It is observed that the average localization error decreases as the percentage of beacon nodes increases in all the algorithms. This is due to the reasons that when there are more beacon nodes in the network, percentages of nodes which have at least four neighbouring beacons becomes greater and localization is done easily and accurately. It is observed that if there is more than 60% of the beacon node, then the nodes localize themselves very accurately. They are very near to the actual coordinates. It is seen from Fig. 4. The localization error for GWO-DV Hop and Weighted GWO-DV Hop reduces from 50 to 35% with the increase in beacon nodes from 20 to 80, whereas Original DV Hop is reduced from 60 to 45%. Thus there is improvement of 10% in accuracy.

Figures 5 and 6 show the impact of total number of nodes and communication radius on average localization error, respectively. If the network size considered is  $100 \times 100 \text{ m}^2$ , then for Fig. 5 where the total nodes are increased from 200 to 450 with beacon percentage

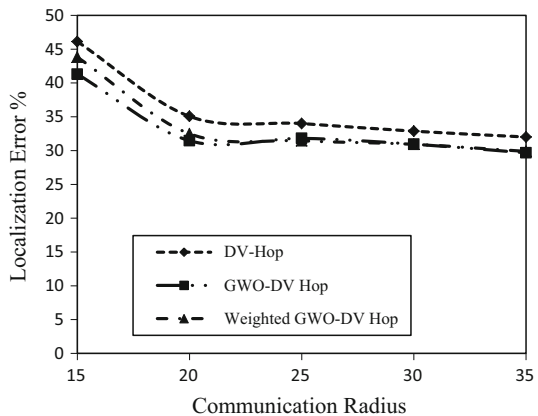
**Fig. 4** Comparison by varying number of beacons (total nodes = 200, border length = 100,  $r = 15$  m)



**Fig. 5** Comparison by varying total number of nodes (beacon nodes = 10% of total nodes, border length = 100,  $r = 15$  m)

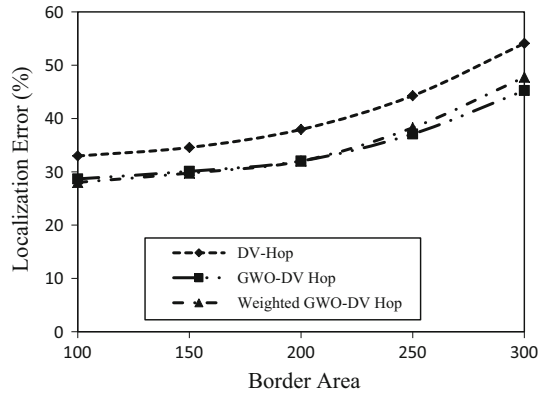


**Fig. 6** Comparison by varying communication radius (total nodes = 200, border length = 100, beacon nodes = 20)

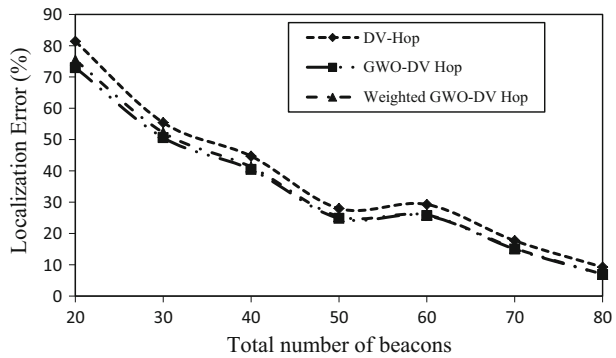


as 10% of total nodes, the localization error is reduced with both proposed algorithms from 42 to 35%. It can be noticed that both the proposed algorithms behave similarly, but better than DV-Hop algorithm. It is also observed that localization accuracy depends upon

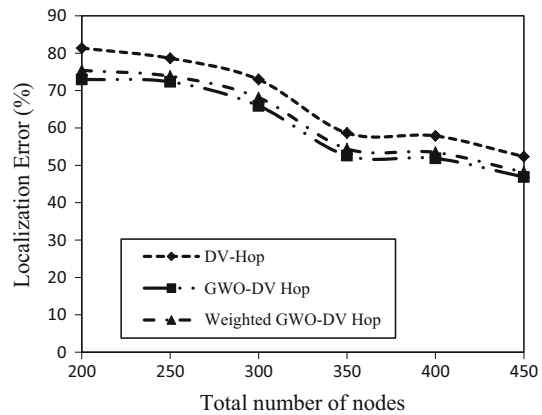
**Fig. 7** Comparison by varying border area (total nodes = 300, beacon nodes = 30, communication radius = 15 m)



**Fig. 8** Comparison by varying total number of beacons (total nodes = 200, border length = 100, communication radius = 20 m)

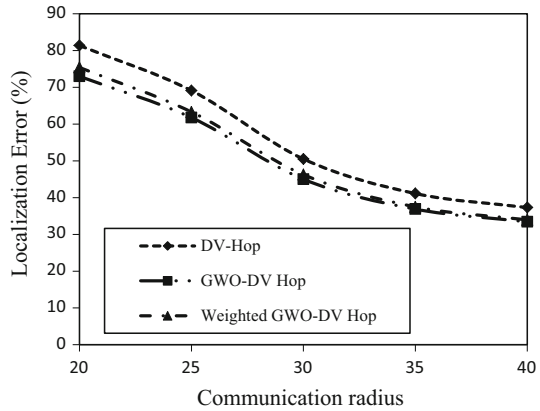


**Fig. 9** Comparison by varying total number of nodes (beacon nodes = 10% of total nodes, border length = 100, communication radius = 20 m)



beacon percentage. There is slight effect on localization accuracy even if total nodes are varied, but the percentage of beacons remains constant. In Fig. 6, communication radius is changed from 15 to 35 in  $100 \times 100 \text{ m}^2$  WSN area with 200 total nodes and 10% beacons. The localization error is reduced by 5% with our proposed algorithms.

**Fig. 10** Comparison by varying communication radius (total nodes = 200, border length = 100, beacon nodes = 20)



**Fig. 11** Comparison by varying border area (total nodes = 300, beacon nodes = 30, communication radius = 20 m)

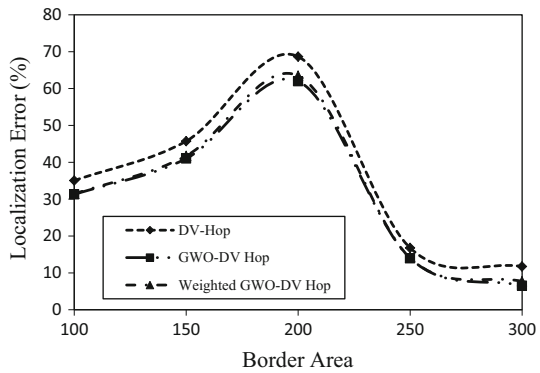


Figure 7 shows the average localization errors of above said algorithms with different border area in the 2D network. The border area is increased from  $100 \times 100$  to  $300 \times 300$  m<sup>2</sup>. It can be seen that with the increase in the border area with the same number of nodes, the node density decreases, thus the localization error increases.

Figures 8 and 9 show the impact on localization error by changing the beacon nodes and total number of nodes, respectively. The beacon nodes are tuned from 20 to 80 for 3D network in Fig. 8 and total nodes are tuned from 200 to 350 for 3D network in Fig. 9 with 10% of beacon nodes. In the both cases, the proposed algorithms outperform the original algorithm.

Figures 10 and 11 show the comparison by varying communication radius and border area. Overall, it is obvious that the GWO-DV Hop performs best with least localization error. The weighted GWO-DV Hop is better than DV-Hop and has slight higher error than GWO-DV Hop algorithm.

### 7.2 Complexity Analysis of Proposed Algorithms

The complexity is considered in terms of communication cost and computation cost [34, 35]. The Communication cost is total number of packets transmitted in the network. There is no change in communication cost for the proposed algorithms.

**Table 4** Computation cost

Phases	DV-Hop	GWO-DV_Hop	Weighted GWO-DV_Hop
1	$O(n^2)$	$O(n^2)$	$O(n^2)$
2	$O(n^2)$	$O(n^2 + mt)$ where $t$ is iterations in GWO	$O(n^2 + mt)$ where $t$ is iterations in GWO
3	$O(n)$	$O(n)$	$O(n)$

The Computation cost that is sum of mathematical operations required to run the algorithms is shown in Table 4.

From Table 4, it is observed that the algorithms give higher accuracy with slight increase in computation cost.

## 8 Conclusions

In order to reduce localization error of DV-Hop localization algorithm due to approximate computation of average hop sizes by each beacon, Grey wolf optimization algorithm is used to improve the estimate of the average hop size. Based on Grey wolf algorithm, two algorithms are proposed. The first algorithm refines the second phase of DV-Hop by obtaining refined average hop size. The second algorithm first applies grey wolf algorithm and then weighted approach is applied to get better approximation of average hop sizes and to consider the impact of every beacon. The simulation results prove that two algorithms perform better than previous DV-Hop. The localization error is reduced by almost 10% by applying grey wolf algorithm in second phase of DV-Hop.

## References

1. Rawat, P., Singh, K. D., Chaouchi, H., & Bonnin, J. M. (2014). Wireless sensor networks: A survey on recent developments and potential synergies. *The Journal of Supercomputing*, 68(1), 1–48.
2. Kaur, A., Kumar, P., & Gupta, G. P. (2017). A weighted centroid localization algorithm for randomly deployed wireless sensor networks. *Journal of King Saud University-Computer and Information Sciences* <https://doi.org/10.1016/j.jksuci.2017.01.007>.
3. Han, G., Xu, H., Duong, T. Q., Jiang, J., & Hara, T. (2013). Localization algorithms of wireless sensor networks: a survey. *Telecommunication Systems*, 52(4), 2419–2436.
4. Zhao, J., Xi, W., He, Y., Liu, Y., Li, X., Mo, L., et al. (2013). Localization of wireless sensor networks in the wild: Pursuit of ranging quality. *IEEE/ACM Transactions on Networking*, 21(1), 311–323.
5. Kunz, T., & Tatham, B. (2012). Localization in wireless sensor networks and anchor placement. *Journal of Sensor and Actuator Networks*, 1(1), 36–58.
6. Girod, L., Bychobvskiy, V., Elson, J., & Estrin, D. (2002). Locating tiny sensors in time and space: a case study. In *Proceedings of the 2002 IEEE international conference on computer design: VLSI in computers and processors*, Los Alamitos (pp. 214–219).
7. Harter, A., Hopper, A., Steggles, P., Ward, A., & Webster, P. (2002). The anatomy of a context-aware application. *Wireless Networks*, 8(2), 187–197.
8. Cheng, X., Thaeler, A., Xue, G., & Chen, D. (2004). TPS: a time-based positioning scheme for outdoor wireless sensor networks. In *Proceedings of the 23rd IEEE annual joint conference of the IEEE computer and communications societies (INFOCOM'04)*, Hong Kong, China, March 2004 (pp. 2685–2696).
9. Niculescu, D., & Nath, B. (2003). Ad hoc positioning system (APS) using AoA. In *Twenty-second annual joint conference of the IEEE computer and communications* (Vol. 3, pp. 1734–1743). IEEE Societies.
10. Bulusu, N., Heidemann, J., & Estrin, D. (2000). GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5), 28–34.

11. Niculescu, D., & Nath, B. (2001). Ad hoc positioning system. In *IEEE on global telecommunications conference* (Vol. 5, pp. 2926–2931).
12. Nagpal, R. (1999). Organizing a global coordinate system from local information on an amorphous computer. A.I. Memo1666, MIT A.I. Laboratory.
13. Shang, Y., & Ruml, W. (2004). Improved MDS-based localization. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'04), Hong Kong, March 2004* (pp. 2640–2651).
14. He, T., Huang, C. D., Blum, B. M., Stankovic, J. A., & Abdelzaher, T. (2003). Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th annual international conference on mobile computing and networking* (pp. 81–95). San Diego, Calif, USA: ACM.
15. Kaur, A., Gupta, G. P., & Kumar, P. (2017). A survey of recent developments in DV-Hop localization techniques for wireless sensor network. *Journal of Telecommunication, Electronic and Computer Engineering*, 9(2), 61–71.
16. Tomic, S., & Mezei, I. (2016). Improvements of DV-Hop localization algorithm for wireless sensor networks. *Telecommunication Systems*, 61(1), 93–106.
17. Song, G., & Tam, D. (2015). Two novel DV-Hop localization algorithms for randomly deployed wireless sensor networks. *International Journal of Distributed Sensor Networks*, 11(7), 187670. <https://doi.org/10.1155/2015/187670>.
18. Zhang, B., Ji, M., & Shan, L. (2012). A weighted centroid localization algorithm based on DV-hop for wireless sensor network. In *Proceedings of the 8th international conference on wireless communications, networking and mobile computing, Shanghai, China* (pp. 1–5).
19. Fang, X. (2015). Improved DV-Hop positioning algorithm based on compensation coefficient. *Journal of Software Engineering*, 9(3), 650–657.
20. Peng, B., & Li, L. (2015). An improved localization algorithm based on genetic algorithm in wireless sensor networks. *Cognitive Neurodynamics*, 9(2), 249–256.
21. Gui, L., Val, T., Wei, A., & Dalce, R. (2015). Improvement of range-free localization technology by a novel DV-hop protocol in wireless sensor networks. *Ad Hoc Networks*, 24, 55–73.
22. Shahzad, F., Sheltami, T. R., & Shakshuki, E. M. (2017). DV-maxHop: A fast and accurate range-free localization algorithm for anisotropic wireless networks. *IEEE Transactions on Mobile Computing*, 16(9), 2494–2505.
23. Wang, F., Wang, C., Wang, Z., & Zhang, X. Y. (2015). A hybrid algorithm of GA+ simplex method in the WSN localization. *International Journal of Distributed Sensor Networks*, 11(7), 731894.
24. Kumar, S., & Lobiyal, D. K. (2016). Novel DV-Hop localization algorithm for wireless sensor networks. *Telecommunication Systems*, 64(3), 509–524.
25. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61.
26. Kaur, A., Kumar, P., & Gupta, G. P. (2016). A novel DV-Hop algorithm based on Gauss–Newton method. In *IEEE international conference on parallel, distributed and grid computing* (pp. 625–629).
27. Keshtgary, M., Fasihy, M., & Ronaghi, Z. (2011). Performance evaluation of hop-based range-free localization methods in wireless sensor networks. *ISRN Communications and Networking*, 2011, 485486. <https://doi.org/10.5402/2011/485486>.
28. Kulaib, A. R., Shubair, R. M., Al-Qutayri, M. A., & Ng, J. W. P. (2015). Improved DV-hop localization using node repositioning and clustering. In *2015 International conference on communications, signal processing, and their applications (ICCSPA)* (pp. 1–6). IEEE.
29. Ping, W. Z., & Chen, X. (2015). Node localization of wireless sensor networks based on DV-hop and Steffensen iterative method. *International Journal of Future Generation Communication and Networking*, 8(2), 1–8.
30. Kumar, S., & Lobiyal, D. K. (2017). Novel DV-Hop localization algorithm for wireless sensor networks. *Telecommunication Systems*, 64(3), 509–524.
31. Cota-Ruiz, J., Rosiles, J. G., Sifuentes, E., & Rivas-Perea, P. (2012). A low-complexity geometric bilateration method for localization in wireless sensor networks and its comparison with least-squares methods. *Sensors*, 12(1), 839–862.
32. Kumar, S., & Lobiyal, D. K. (2013). Improvement over DV-Hop localization algorithm for wireless sensor networks. *World Academy of Science, Engineering and Technology*, 76(4), 282–293.
33. Ahmad, T., Li, X. J., & Seet, B. C. (2017). Parametric loop division for 3D localization in wireless sensor networks. *Sensors*, 17(7), 1697. <https://doi.org/10.3390/s17071697>.
34. Keshtgary, M., Fasihy, M., & Ronaghi, Z. (2011). Performance evaluation of hop-based range-free localization methods in wireless sensor networks. *ISRN Communications and Networking*, 2011. <https://doi.org/10.5402/2011/485486>.
35. Singh, M., & Khilar, P. M. (2016). An analytical geometric range free localization scheme based on mobile beacon points in wireless sensor network. *Wireless Networks*, 22(8), 2537–2550.



**Amanpreet Kaur** is pursuing Ph.D. in Computer Science and IT from Jaypee Institute of Information Technology, Noida. She has done MTech from NIT, Jalandhar. Her research areas are Wireless sensor networks, Information Security, Performance Analysis.



**Prof. Padam Kumar** is currently working as Dean-RID in Jaypee Institute of Information Technology, Noida. Previously, he worked as Professor and Head of Department of Computer Science and Engineering, IIT Roorkee. He is a member of IEEE. His research areas are Declarative languages, Functional Programming, Parallel and Distributed Systems, Multiprocessor Scheduling, Real-time Multiprocessor systems, Fault-tolerant systems, Computer Surveillance, Cloud Security.



**Dr. Govind P. Gupta** is currently working as Assistant Professor in National Institute of Technology, Raipur. He has done Ph.D. from IIT, Roorkee. His area of interests are Computer Networking, Distributed Algorithms design for Wireless Sensor Networks, Performance Analysis, Big Data Processing, Parallel and Distributed Computing, Design & Analysis of Algorithms.