Abstract—Wireless Sensor Networks (WSNs) applications have attracted attention in Internet of Things (IoT) as a novel networking paradigm consisting of billions of small sensor nodes. These sensors collect environmental information and communicate with each other to provide solutions for real-time IoT applications’ requirements. Since the majority of applications require location-based services, it is necessary to improve the accuracy of localization algorithms. DV-Hop is one of the most attractive range-free localization algorithms in wireless sensor networks and several works have been undertaken to improve its accuracy, however, since sensor nodes have limited power resources, the energy consumption of nodes should also be considered. In this paper, we propose a method based on DV-Hop to improve both accuracy and power consumption. Each unknown node calculates the Hopsize of each anchor node according to the limited information it has from the network topology; therefore there is no need to broadcast the Hopsize from anchor nodes, and, in this way, energy can be saved. In the next step, we use Shuffled Frog Leaping Algorithm (SFLA) as an evolutionary algorithm to improve the accuracy of estimated Hopsizes and a hybrid Genetic-PSO algorithm is applied to the third step of DV-Hop to achieve more accurate values for unknown nodes’ positions. Simulation results show that our proposed method decreases the localization error significantly by jointly considering the energy consumption of sensors and is overall 44% more accurate than DV-Hop.

Index Terms—Wireless Sensor Networks, Internet of Things, Localization, DV-Hop, Hopsize, Shuffled Frog Leaping Algorithm, Evolutionary

I. INTRODUCTION

The basic idea of the Internet of Things (IoT) is to have a connected model of smart sensors which are collaborating with each other to provide a new category of services. These sensors consist of a transceiver, limited computational processor and power supply, can form wireless sensor networks (WSNs) to collect environmental information and evaluate or process them based on the applications’ requirements. The location information of sensors play an important role in several use case scenarios such as health care, smart house, industrial production and military applications [1]–[3]. Global Positioning System (GPS) [4] is the easiest way to determine a node’s location. However, due to the limitation in power resources of sensor nodes, using GPS for devices is detrimental because of high energy consumption. Therefore, different localization methods have been proposed which are categorized into two distinct groups, namely range-based and range-free methods. There are some nodes available in the network using GPS to localize themselves accurately, known as anchor nodes, while other nodes which estimate their positions are called unknown nodes [5].

Range-based algorithms determine nodes’ location using end-to-end distance, angle information and lateration method. RSSI [6], Angle of Arrival (AoA) [7] and Time of Arrival (ToA) [8] are some examples of range-based methods. On the other hand, range-free methods such as Distance Vector-Hop (DV-Hop) [9], Centroid [10] and Approximate Point In Triangulation (APIT) [11] are only based on the connectivity among nodes and depend on the network topology. Although these methods are less accurate, they do not need extra devices. Therefore, they are more cost beneficial and have lower power consumption. Among range-free methods, DV-Hop has the lowest accuracy. However, since it does not require at least three anchor nodes in the neighborhood of each unknown node to estimate their positions, its complexity is also low. Considering this advantage, DV-Hop has attracted researchers’ attention to improve its accuracy. Recently, many scholars have tried to reduce localization error by using evolutionary algorithms [12]–[14]. However, the majority have just considered the localization accuracy without paying attention to the energy consumption of nodes [15].

In this paper, we propose a method based on DV-Hop to enhance both localization accuracy and power consumption of nodes using evolutionary algorithms. In DV-Hop, each anchor node broadcasts information in two steps: in the first step, it sends the information about its position and in the second step, it sends the information of its Hopsize. In both steps, all nodes in the network send and receive information. Therefore, most of the nodes’ energy is used to transmit data. Our proposed method extends on our previous work towards reaching a more accurate and energy-efficient localization algorithm [12]. However, the main significance of the present work is reduction of the power consumption in the network while improving accuracy by reducing the number of sending/receiving operations. In order to achieve this, we change the way of calculating the Hopsize of anchor nodes such that each unknown node estimates the Hopsize according to the information it receives in the first step from anchor
nodes. Therefore, there is no need to broadcast the Hopsize of anchor nodes in the second step, and thus the energy can be saved. In order to decrease the error of the calculated Hopsize, the Shuffled Frog Leaping Algorithm (SFLA) is employed. Our proposed algorithm can achieve lower error and lower power consumption due to the reduction in the number of broadcasts.

The remaining of this paper is structured as follows. In the second section, Basic DV-Hop and the used evolutionary algorithms are explained. In Section III, our proposed scheme and the calculation method of the localization error are detailed. In Section IV, simulation results are provided, and finally in the last section, the conclusion and future works are presented.

II. PRINCIPLES OF DV-HOP AND OPTIMIZATION ALGORITHMS

A. DV-Hop Algorithm

The DV-Hop algorithm, which was proposed by D. Niculescu and B. Nath in 2002 [9], is one of the most promising range-free methods. In DV-Hop, there are two kind of nodes, namely anchor and unknown nodes. The anchor nodes are equipped by GPS and are aware of their locations, and the location of each unknown node is estimated in three steps as follows:

1) Each anchor node broadcasts a packet including information about its position and a hop-count, which is initialized to zero. Each neighbor node receives these information, saves them in a table and increments the hop-count number, then broadcasts the updated information. Note that if a node receives another message from the same anchor node but with a higher hop-count, the node ignores the message. At the end, the hop-counts among all anchor nodes will be determined.

2) The Hopsize of anchor node $i$ is calculated using equation (1) as the average distance of one hop:

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

where $(x_i, y_i)$ and $(x_j, y_j)$ are the coordinates of anchor nodes $i$ and $j$, respectively, $h_{ij}$ is the hop-count between these anchor nodes and $N$ is the number of total anchor nodes in the network. The anchor nodes again broadcast these information, and each unknown node only accepts the Hopsize of the nearest anchor node and ignores others.

3) Unknown nodes use these Hopsizes to convert hop-counts to their physical distances to anchor nodes:

$$d_{ij} = \text{Hopsize}_i \times h_{ij},$$

where $h_{ij}$ is the minimum hop-count between the anchor node $i$ and the unknown node $j$. By determining these distances and solving the resulting system of equations (3), the estimated position of the unknown node can be calculated using the least squares method:

$$\begin{align*}
\sqrt{(x - x_1)^2 + (y - y_1)^2} &= d_1 \\
\sqrt{(x - x_2)^2 + (y - y_2)^2} &= d_2 \\
&\vdots \\
\sqrt{(x - x_N)^2 + (y - y_N)^2} &= d_N
\end{align*}$$

B. Shuffled Frog Leaping Algorithm

The Shuffled Frog Leaping Algorithm (SFLA) is based on the interaction among individuals, which are divided into different groups [16]. In SFLA, a population of $N$ virtual frogs is generated and divided into $M$ groups called memeplexes while sorted in ascending order according to a fitness function. This categorization happens in the way that frogs from the first to the $(m)$-th rank are placed in the $M$ memeplexes respectively. This process continues until all frogs are grouped in the memeplexes. The worst frog of each memeplex is replaced with a new one according to equation (4) if the new frog has a better evaluation in the fitness function:

$$X_{new} = X_w + R(X_b - X_w),$$

where $X_b$ and $X_w$ are the best and the worst frogs of the memeplex $k$, respectively, and $R$ is a random number ranging from 0 to 1. If the new frog is not better than the previous one, $X_g$ (the global best frog) is placed instead of the $X_b$ in equation (4). Again, if the new frog does not meet the condition, the worst frog of the memeplex $k$ is replaced by the random frog. After replacing all worst frogs from all memeplexes, they are scrambled and reevaluated according to the fitness function to be grouped in $M$ memeplexes again. The whole process continues until reaching to the desired condition or predetermined number of iterations.

C. Hybrid Genetic-PSO Algorithm

The Genetic algorithm (GA), which was firstly proposed by J. Holland [17], includes some chromosomes as candidate solutions for a particular problem with parameters called genes. In our localization problem, each chromosome has two genes for its $x$- and $y$- coordinates to represent a point in the search space. The chromosomes are sorted from the best to the worst based on their fitness function values. Selection, crossover and mutation are three basic operators used by GAs to control the population of chromosomes towards the optimal point. In the next step, offspring is produced from each two chromosomes, which are chosen by a selection mechanism to create the next generation. By using crossover operators, offspring inherits features from both parents. Furthermore, GA uses mutation operators to create new characteristics in the chromosomes as a crucial point in evolution. This procedure continues for a predetermined number of iterations.

Particle Swarm Optimization (PSO), which is introduced by J. Kennedy [18], includes mobile particles with the main procedure of updating their velocity and location towards reaching an optimum value. The best solution experienced by
each particle (local best) and the best solution experienced by all particles (global best) are two parameters which influence movements of particles.

In Hybrid Genetic-PSO algorithm [19], a population of the best individuals with the highest fitness values are chosen and optimized by the PSO algorithm to reduce the convergence time of our proposed method. The population is then transferred to the next generation. In addition, to reduce the complexity of the hybrid algorithm, the velocity and location of each particle are updated based on equations (5) and (6):

\[
v_i(t + 1) = c_1 r_1 [g(t) - x_i(t)],
\]

\[
x_i(t + 1) = x_i(t) + v_i(t + 1),
\]

where \(v_i\) and \(x_i\) are the current velocity and position of particle \(i\), \(c_1\) denotes a learning factor between 0 and 2, \(r_1\) is a random number between 0 and 1, \(g\) is the global best in group. In this way, the computation is simple and combining two algorithms adds only a slight computational load.

### III. PROPOSED METHOD

The first step of our proposed algorithm is similar to that of DV-Hop. The idea of calculating the Hopsize of anchor nodes in our second step is inspired by [15] where unknown nodes can obtain this value by only using the information received from the first step and the communication radius \(R\), considering the fact that the maximum number of Hopsize can be equal to \(R\). In this way, there is no need for broadcasting the Hopsizes of anchor nodes in the second step, and thus the energy that nodes spend on communications can be significantly reduced. The number of hops between anchor nodes \(i\) and \(j\) can be calculated by

\[
h_{ij} = \lfloor \frac{d_{ij}}{R} \rfloor, \tag{7}
\]

where \(R\) is the communication radius and \(d_{ij}\) is the distance between anchor nodes \(i\) and \(j\).

Then, the unknown node obtains the Hopsize of each anchor by using their distance and hop-counts information and takes the average due to the effects of other anchor nodes on this value:

\[
\text{Hopsize}_{i} = \frac{\sum_{j=1,j \neq i}^{N} \frac{d_{ik}}{h_{ij}}}{N}. \tag{8}
\]

The assumption that all anchor nodes have the same effect on Hopsize calculation is not warranted, and as it has been proved in [20], the anchor nodes closer to an unknown node should have a bigger effect in obtaining a more accurate position value. To address this issue, in this paper, we consider a weight for each anchor node so that anchor nodes with higher impact on the unknown node position estimation have higher weights. Thus, the Hopsize can be calculated more accurately and the Hopsize error will be reduced. The assigned weight is calculated as follows:

\[
w_{ki} = \frac{1}{h_{ik} + h_{jk}}, \tag{9}
\]

where \(h_{ik}\) and \(h_{jk}\) are the number of hops from the unknown node \(k\) to the anchor nodes \(i\) and \(j\), respectively. Thus, the modified Hopsize equation becomes:

\[
\text{Hopsize}_{k,i} = \frac{\sum_{j=1,j \neq k}^{N} (w_{ki} \times d_{ij})}{N}. \tag{10}
\]

Suppose that \(\text{Hopsize}_{i,k}\) is the average Hopsize of anchor node \(i\), \(d_{ij}\) and \(h_{ij}\) are the actual distance and the number of hops between anchor nodes \(i\) and \(j\), respectively, and \(e_i\) is the Hopsize error of anchor node \(i\). Then, we have the following equation:

\[
d_{ij} = \text{Hopsize}_{i,k} \times h_{ij} + e_i. \tag{11}
\]

In the next step, we apply the SFSLA as the optimization method in order to reduce the error in equation (11). So, our fitness function can be expressed as follows:

\[
f(\text{Hopsize}_i) = \frac{1}{N - 1} \sum_{j=1,j \neq i}^{N} e_i^2. \tag{12}
\]

Here, frogs are representative forHopsize, so each individual only has one parameter. The initialized population of frogs are generated randomly between 0 and \(\max(d_{ij})\). At the end, the best frog is our new Hopsize. The Hopsize, used by unknown node \(k\) to determine its estimated distances to anchor nodes, is calculated as the average Hopsizes of all anchor nodes:

\[
\text{Hopsize}_{k,\text{avg}} = \frac{\sum_{i=1}^{N} \text{Hopsize}_{k,i}}{N}. \tag{13}
\]

To achieve more accurate estimation of the unknown node’s position, the difference between the estimated and the real distances from the unknown node to the anchor nodes should also be minimized in the next step. To this end, instead of solving the system of equations shown in (3), we find the minimum value of the following fitness function:

\[
F(x, y) = \min \left( \sum_{i=1}^{N} \sqrt{(x - x_i)^2 + (y - y_i)^2 - d_i^2} \right), \tag{14}
\]

where \(d_i\) is the distance between the unknown node and the anchor node \(i\), \((x_i, y_i)\) is the coordinate of anchor node \(i\), and \(N\) is the total number of anchor nodes. Considering the fact that GA always aims at maximizing the fitness function, we inverse the fitness function according to (12):

\[
\text{fitness}(x, y) = \frac{\alpha}{1 + \beta F(x, y)}, \tag{15}
\]

where \(\alpha\) and \(\beta\) are real positive numbers. Then, we apply the hybrid Genetic-PSO algorithm to this fitness function.

In the hybrid algorithm, the chromosomes represent the unknown nodes’ position. Therefore, each chromosome has two parameters, namely \(x\) and \(y\) as values of position. The initialized population is randomly generated in the red area in Fig.1. This region includes the common area of three squares.
with centers located on the three nearest anchor nodes, which have the highest accuracy in the estimation of the unknown node’s position due to the low number of hops between them and the unknown node. The authors in [21] used the communication radius, however, the size of this area is a key factor to speed up the convergence of the algorithm. Therefore, we modify this search space by using the calculated Hopsizes in the second step.

Therefore, the lower and upper coordinates of the common region around the unknown node \( n \) are defined as follows:

\[
\begin{align*}
\max_{i=1,2,3} (x_i - \text{Hopsize}_{i,n} \times H_i) & \leq x_n \\
\min_{i=1,2,3} (x_i + \text{Hopsize}_{i,n} \times H_i) & \leq x_n \\
\max_{i=1,2,3} (y_i - \text{Hopsize}_{i,n} \times H_i) & \leq y_n \\
\min_{i=1,2,3} (y_i + \text{Hopsize}_{i,n} \times H_i) & \leq y_n,
\end{align*}
\]

(16)

where \((x_n, y_n)\) are the coordinates of the unknown node, \( H_i \) is the number of its hops to the anchor node \( i \), and \( \text{Hopsize}_{i,n} \) is the estimated distance of one hop for anchor nodes according to SFLA in the previous step.

\[
\text{Error} = \frac{1}{M} \sum_{i=1}^{M} \sqrt{(x_{est,i} - x_i)^2 + (y_{est,i} - y_i)^2},
\]

(17)

where \((x_{est,i}, y_{est,i})\) and \((x_i, y_i)\) are the estimated and actual coordinates of unknown node \( i \) respectively and \( M \) is the number of unknown nodes. In the following, we analyze the results with different focus points.

- **The localization error of unknown nodes:**
  The error of each unknown node’s location for a single simulation run is shown in Fig. 2. Simulations are conducted for total number of 100 sensor nodes with 10 anchor nodes and communication radius of 40m for each node. As it can be observed, the localization accuracy of our proposed method is better compared to DV-Hop and to the algorithm in [12] due to the Hopsize modification in the second step.

\[
\text{Number of Hybrid algorithm iterations} = 10
\]

\[
\text{Number of PSO iterations} = 1
\]

### TABLE I

**Simulation parameters**

| Parameter                          | Value          |
|------------------------------------|----------------|
| Network size                       | \( 100 \times 100 \text{ m}^2 \) |
| Total number of nodes              | 100            |
| Number of frogs                    | 40             |
| Number of memplexes                | 4              |
| Number of iterations within each memplex | 10             |
| Number of SFLA iterations          | 20             |
| Number of chromosomes              | 10             |
| Number of Hybrid algorithm iterations | 10          |
| Mutation probability               | 0.2            |
| Crossover probability              | 0.4            |
| Number of particles                | 12             |

Fig. 1. Population common region for the hybrid Genetic-PSO algorithm.

Fig. 2. Localization error of each unknown node.

- **Effect of varying number of anchor nodes on localization error:**
  Fig. 3 shows the results of localization errors for the original DV-Hop, the algorithm in [12] and our proposed method with varying number of anchor nodes in network ranging from 10 to 40 with communication radius of 40m. 100 independent simulation runs have been done to obtain reliable results. It can be seen that as the number of anchor nodes increases from 10 to 40, the
localization error decreases and our proposed method has the lowest error among all competitors. The hop-counts between anchor nodes and unknown nodes are reduced by increasing the number of anchor nodes in the network, consequently the error in Hopsize calculation is reduced, which leads to reduction of accumulated error of estimated distances.

- **Effect of different communication radius on localization accuracy:**
  Fig. 4 shows the error results for a total number of 10 anchor nodes and the communication radius ranging from 30m to 50m with 100 independent simulation runs to obtain reliable results. With increasing communication radius, the value for one hop is also increased; consequently, the accumulated error of estimated distances will affect the localization error. As it is shown in the Fig. 4, our proposed method has the lowest error compared to the others. The reason for this is our Hopsize calculation method which incorporates relationships between unknown nodes and anchor nodes, while other methods only calculate it by using anchor nodes’ information.

- **Comparison of energy consumption:**
  Sensors spend more energy on data transmission than on computation, so we only consider the energy for the transmission part in our comparison. In the DV-Hop algorithm, information are broadcasted in two steps, first for sending the anchor nodes’ positions and then after calculating the Hopsize of anchor nodes. The broadcasting procedure in [12] is same as DV-Hop. Therefore, we have just shown the results for DV-Hop and our proposed algorithm. In this work, one step of broadcasting is eliminated since unknown nodes calculate Hopsize by the information they receive in the first step. Fig. 5 shows the total amounts of packet transmissions within the network for 10 anchor nodes and communication radius of 40m.

- **Convergence speed:**
  One of the key factors for evaluating the efficiency of evolutionary algorithms is their convergence speed and the time needed to reach the optimum value. Several parameters such as population size, population space, combination and mutation ratios can influence the speed of the evolutionary algorithms convergence. Fig. 6 shows the convergence speed of our proposed method compared to the algorithm presented in [12]. As it can be seen, the proposed algorithm generally can reach the optimum value in a low number of iterations which means the algorithm doesn’t need much time for computation and this can save the energy of sensors while using evolutionary methods to reach the more accurate localization parameters. The proposed algorithm is also faster than the algorithm proposed in [12]. The reason is the smaller population region as the result of our Hopsize calculation.
method and more accurate distance per hops.

![Fig. 6. Convergence speed.](image)

V. CONCLUSION AND FUTURE WORKS

Recent IoT applications need exact locations of sensors. In this paper, we proposed a method to jointly optimize the accuracy and the energy consumption of DV-Hop algorithm. In our method, each unknown node can calculate Hopsizes of anchor nodes, which eliminates the broadcasting procedure in the second step of DV-Hop. SFLA then is used to improve the error of estimated Hopsizes. In the last step, we obtained more accurate values for the location of nodes using a hybrid Genetic-PSO algorithm. Simulation results have shown that our proposed method has the lowest localization error in comparison with the original DV-Hop and the algorithm presented in [12]. The cost for this is a negligible increase in the computation time.

For future, we plan to improve the localization accuracy by changing the population region of evolutionary algorithms. Furthermore, the fact that future applications will require 3D localization and the mobility of sensors should be also considered.

ACKNOWLEDGMENT

This research has been supported by European Union’s H2020 research and innovation program under grant agreement H2020-MCSA-ITN-2016-SECRET 722424 [22] and the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany’s Excellence Strategy EXC2050/1 Project ID 390696704 Cluster of Excellence Centre for Tactile Internet with Human-in-the-Loop (CeTI) of Technische Universität Dresden. The authors alone are responsible for the content of the paper.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” IEEE communications surveys & tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] R. C. Shiit, S. Sharma, D. Puthal, and A. Y. Zomaya, “Location of things (lot): A review and taxonomy of sensors localization in iot infrastructure,” IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2028–2061, 2018.

[3] G. Sharma and A. Komai, “Improved dv-hop localization algorithm using teaching learning based optimization for wireless sensor networks,” Telecommunication Systems, vol. 67, no. 2, pp. 163–178, 2018.

[4] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, Global positioning system: theory and practice. Springer Science & Business Media, 2012.

[5] A. Mesmoudi, M. Feham, and N. Labnouati, “Wireless sensor networks localization algorithms: a comprehensive survey,” arXiv preprint arXiv:1312.4082, 2013.

[6] L. Girod, V. Bichkovsky, J. Elson, and D. Estrin, “Locating tiny sensors in time and space: A case study,” in Proceedings. IEEE International Conference on Computer Design: VLSI in Computers and Processors. IEEE, 2002, pp. 214–219.

[7] M. Mehrabi, H. Taheri, and P. Taghirad, “An improved dv-hop localization algorithm based on evolutionary algorithms,” Telecommunication Systems, vol. 64, no. 4, pp. 639–647, 2017.

[8] J. Rodriguez, A. Radwan, C. Barbosa, F. H. Fitzek, R. A. Abd-Alhameed, J. Noras, S. M. Jones, J. Politis, P. Galiotis, G. Schulte et al., “Secret–secure network coding for reduced energy next generation mobile small cells: A European training network in wireless communications and networking for 5G,” in 2017 Internet Technologies and Applications (ITA). IEEE, 2017, pp. 329–333.