Hindawi
Journal of Sensors
Volume 2022, Article ID 7026728, 18 pages
https://doi.org/10.1155/2022/7026728

Research Article

Node Localization Algorithm Based on Modified Archimedes Optimization Algorithm in Wireless Sensor Networks

Mangmang Cheng,[1] **Tao Qin**,[1] **and Jing Yang**[1,2]

[1]*Electrical Engineering College, Guizhou University, Guiyang 550025, China*
[2]*Guizhou Provincial Key Laboratory of Internet + Intelligent Manufacturing, Guiyang 550025, China*

Correspondence should be addressed to Jing Yang; jyang7@gzu.edu.cn

Received 19 November 2021; Accepted 11 May 2022; Published 7 June 2022

Academic Editor: Christos Riziotis

Node localization information plays an important role in wireless sensor networks (WSNs). To solve the problem of low localization accuracy of distance vector hop (DV-Hop) localization algorithm in wireless sensor networks, an improved localization algorithm called MAOADV-Hop based on the modified Archimedes optimization algorithm (MAOA) and DV-Hop is proposed, which can achieve the balance between the localization speed and the localization precision. Firstly, tent chaotic mapping and particle swarm optimization (PSO) algorithm are introduced into Archimedes optimization algorithm to improve the initial population diversity and change the update rules of density and volume, which improve the global convergence ability and convergence speed of the algorithm. Secondly, the MAOA is used to replace the least square part of the DV-Hop localization algorithm to improve the localization accuracy of the algorithm. Finally, MAOADV-Hop is verified through four different network environments and compared with DE_DV-Hop, BOA_DV-Hop, and DV-Hop. The simulation results show that the localization speed of the proposed approach is faster than that of DE_DV-Hop and BOA_DV-Hop, and the localization error is less than that of DV-Hop, DE_DV-Hop, and BOA_DV-Hop.

1. Introduction

Wireless sensor networks (WSNs) consist of a number of static or mobile sensors in self-organizing and multihop manner, aimed at sending the information detected and processed by the sensor nodes in the coverage area of the network to the users [1, 2]. WSNs integrates MEMS, sensor technology with network communication technology [3] and is widely used in agriculture [4], military [5], environmental protection [6], intelligent transportation [7, 8], and other fields. It has been the focus and highlight of international competition because of the focuses of Industry and Academia [9, 10].

Because the location information of nodes plays an important role in the working process of WSNs [11], localization is an indispensable basic technology [12]. Although the current GPS localization system can accurately locate the target, it is difficult to use satellite positioning information for accurately locating the target in some special places [13, 14]. Therefore, it is very meaningful and challenging work to study the accurate localization algorithm for WSNs.

The rest of this paper is organized as follows: In Section 2, we describe the localization algorithms in WSNs and the main contributions of our work. In Section 3, we improve the Archimedes optimization algorithm (AOA) and compared with the other algorithms. In Section 4, the MAOADV-Hop localization algorithm is proposed and the the experimental simulation comparison is carried out to verify the performance of the localization algorithm. Finally, we summarize the work of this paper and describe the future work in Section 5.

2. Literature Review

Node localization algorithms in WSNs are mainly divided into range-based localization algorithm and range-free localization algorithm [15]. The range-based localization algorithm mainly included RSSI [16], TOA [17], TDOA [18], and angle of arrival (AOA) [19]. The range-free localization algorithm included centroid [20], weighted centroid, DV-Hop, amorphous [21], and APIT localization algorithm [22]. The position accuracy of range-free localization algorithm can satisfy most of the needs, which is popular among users [23].

The DV-Hop localization algorithm is one of the most famous range-free localization algorithms in WSNs [24–26] due to its high robustness and simplicity. But, the DV-Hop algorithm has a lower localization accuracy in complex environment. Thus, many improved DV-Hop algorithms have been proposed in recent years.

Messous et al. proposed an improved recursive DV-Hop localization algorithm for randomly deployed wireless sensor networks, which uses an optimization formula to calculate the average hop count of anchor nodes to obtain better localization accuracy [27]. Shi et al. proposed an improved DV-Hop scheme based on path matching and particle swarm optimization algorithm, which uses an improved particle swarm optimization algorithm to optimize the location of unknown nodes [28]. Han et al. proposed an improved localization algorithm based on the improved DV-Hop and differential evolution (DE) algorithm in 2020 [29]. Huang and Zhang proposed a weighted DV-Hop localization algorithm for wireless sensor networks based on DE algorithm [30]. Lei et al. proposed DV-Hop localization based on the improved sparrow search algorithm (SSA) in wireless sensor networks in 2020, using the double communication radius method to improve DV-Hop and the improved sparrow search algorithm to estimate the location of nodes [31]. Huang et al. proposed a three-dimensional localization algorithm for WSNs based on improved A∗ and DV-Hop algorithms [32]. Han et al. proposed a multitarget vector hopping localization algorithm based on differential evolution quantum particle swarm optimization [33].

AOA is an algorithm with fast convergence speed and good global convergence ability [34], but its performance of the global convergence speed can be improved. Thus, a modified AOA (MAOA) with better global convergence ability is proposed. And the MAOA is used to improve the performance of the DV-Hop algorithm, that is, the MAOADV-Hop algorithm.

The main contributions of our work in this paper can be summarized as follows.

The Archimedes algorithm is optimized. Firstly, tent chaotic map is introduced into AOA to increase the diversity of the initial population of the algorithm. Secondly, the concepts of social learning and individual cognition in PSO are introduced to update the iterative equations of density and volume in AOA. It accelerates the convergence of density and volume to enhance the convergence speed of the algorithm.

MAOA is tested on several test functions and compared with AOA, DE, butterfly optimization algorithm (BOA) [35], PSO, and marine predator algorithm (MPA). The performance of the improved MAOA is verified.

The paper proposed a WSN localization algorithm based on MAOA and the DV-Hop algorithm (MAOADV-Hop). Compared with DV-Hop, DE_DV-Hop, and BOA_DV-Hop, the MAOADV-Hop has better convergence rate than that of DE_DV-Hop and BOA_DV-Hop and has better localization accuracy than that of the DV-Hop and BOA_DV-Hop.

3. AOA and MAOA

In this section, Section 3.1 introduces the entire optimization process of AOA. Section 3.2 improves AOA by using the tent chaotic map and the concepts of social learning and individual cognition in PSO, and MAOA is proposed. Section 3.3 verifies the performance of MAOA by comparing with five swarm intelligence optimization algorithms.

3.1. Archimedes Optimization Algorithm

3.1. Archimedes Optimization Algorithm. AOA is a metaheuristic algorithm inspired by Archimedes' principle. Like other population-based metaheuristic algorithms, AOA begins its searching process through an initial population with random volume, density, and acceleration. The steps of the algorithm are as follows.

Step 1. Initialize population position, volume, density, and acceleration using

$$\begin{aligned} X_i &= \mathrm{lb}_i + \mathrm{rand} \times (\mathrm{ub}_i - \mathrm{lb}_i)\ ; i = 1, 2, \cdots, N, \\ \mathrm{acc}_i &= \mathrm{lb}_i + \mathrm{rand} \times (\mathrm{ub}_i - \mathrm{lb}_i)\ ; i = 1, 2, \cdots, N, \\ \mathrm{den}_i &= \mathrm{rand}\ (N, D), \\ \mathrm{vol}_i &= \mathrm{rand}\ (N, D), \end{aligned} \tag{1}$$

where X_i denotes the ith object in N population. N and D, respectively, denote population number and dimensions of the search space. ub_i and lb_i are the lower and upper bounds of the search space, respectively. vol_i, den_i, and acc_i denote volume, density, and acceleration of the ith object, respectively. $\mathrm{rand}\ (N, D)$ is a $N \times D$ dimensional matrix, which can be randomly calculated by the system function. Then, the individual X_{best} with the best fitness value and the corresponding $\mathrm{acc}_{\mathrm{best}}$, $\mathrm{den}_{\mathrm{best}}$, and $\mathrm{vol}_{\mathrm{best}}$ were selected.

Step 2. Update the density and volume of $(t+1)$th iteration of ith object using

$$\begin{aligned} \mathrm{den}_i^{t+1} &= \mathrm{den}_i^t + \mathrm{rand} \times \left(\mathrm{den}_{\mathrm{best}} - \mathrm{den}_i^t\right), \\ \mathrm{vol}_i^{t+1} &= \mathrm{vol}_i^t + \mathrm{rand} \times \left(\mathrm{vol}_{\mathrm{best}} - \mathrm{vol}_i^t\right), \end{aligned} \tag{2}$$

where $\mathrm{den}_{\mathrm{best}}$ and $\mathrm{vol}_{\mathrm{best}}$, respectively, are the global optimal value of density and volume so far.

Step 3. Calculate the parameter TF and the density decline factor d, which are used to balance the local convergence ability and global convergence ability of MAOA.

$$\mathrm{TF} = \exp\left(\frac{t - t_{\max}}{t_{\max}}\right), \quad (3)$$

where $t_{\max}$ is max-iterations and t is the current iterations. TF increases with the number of iterations until TF = 1.

$$d^{t+1} = \exp\left(\frac{t_{\max} - t}{t_{\max}}\right) - \left(\frac{t}{t_{\max}}\right), \quad (4)$$

where d decreases as the number of iterations increases, and the search is transferred to the bounded region that has been identified.

Step 4. If $\mathrm{TF} \le 0.5$, the phase is exploration phase and collision between objects occurs. Update acceleration using

$$\begin{aligned} \mathrm{acc}_i^{t+1} &= \frac{\mathrm{den}_{\mathrm{mr}} + \mathrm{vol}_{\mathrm{mr}} \times \mathrm{acc}_{\mathrm{mr}}}{\mathrm{den}_i^{t+1} + \mathrm{vol}_i^{t+1}}, \\ \mathrm{mr} &= \mathrm{rand}, \end{aligned} \quad (5)$$

where acc_i^{t+1}, den_i^{t+1}, and vol_i^{t+1} denote the acceleration, density, and volume of the ith individual in the $(t+1)$th iteration, respectively. $\mathrm{acc}_{\mathrm{mr}}$, $\mathrm{den}_{\mathrm{mr}}$, and $\mathrm{vol}_{\mathrm{mr}}$ denote the acceleration, density, and volume of random individuals, respectively.

If $\mathrm{TF} > 0.5$, the phase is exploitation phase and no collision between objects. Update acceleration using

$$\mathrm{acc}_i^{t+1} = \frac{\mathrm{den}_{\mathrm{best}} + \mathrm{vol}_{\mathrm{best}} \times \mathrm{acc}_{\mathrm{best}}}{\mathrm{den}_i^{t+1} + \mathrm{vol}_i^{t+1}}. \quad (6)$$

Then, normalize the acceleration using

$$\mathrm{acc}_{i,\mathrm{norm}}^{t+1} = u \times \frac{\mathrm{acc}_i^{t+1} - \min(\mathrm{acc})}{\max(\mathrm{acc}) - \min(\mathrm{acc})} + l, \quad (7)$$

where u and l are the range of normalization and set to 0.9 and 0.1, respectively. $\mathrm{acc}_{i,\mathrm{norm}}^{t+1}$ denotes the percentage of steps that each agent will change. If the object i is far away from global optimum, the $\mathrm{acc}_{i,\mathrm{norm}}^{t+1}$ value will be high which means that the object is in the exploration phase.

Step 5. If $\mathrm{TF} \le 0.5$, update the population X position using

$$X_i^{t+1} = X_i^t + C_1 \times \mathrm{rand} \times \mathrm{acc}_{i,\mathrm{norm}}^{t+1} \times d \times \left(X_{\mathrm{rand}} - X_i^t\right), \quad (8)$$

where C_1 is a constant equal to 2. Otherwise, if $\mathrm{TF} > 0.5$, update the population X position using

$$X_i^{t+1} = X_{\mathrm{best}}^t + F \times C_2 \times \mathrm{rand} \times \mathrm{acc}_{i,\mathrm{norm}}^{t+1} \times d \times \left(T \times X_{\mathrm{best}} - X_i^t\right), \quad (9)$$

where C_1 is a constant equal to 6. $T = C_3 \times TF$, T increases with time. F is a parameter that changes the direction of motion, calculated using

$$\begin{aligned} P &= 2 \times \mathrm{rand} - C_4, \\ F &= \begin{cases} +1, & \text{if } P \le 0.5, \\ -1, & \text{if } P > 0.5, \end{cases} \end{aligned} \quad (10)$$

where C_3 and C_4 are used to balance the movement direction of the population to adjust the ability of the algorithm to jump out of the local optimization.

Step 6. Evaluation. Select the individuals with the best fitness and their acceleration, density, and volume based on the updated population. Then, the algorithm proceeds to the next iteration until the iteration reach the max-iterations.

3.2. The Improvement of Archimedes Optimization Algorithm

3.2.1. Tent Chaotic Mapping. Tent chaotic map can improve the population diversity to strength the global search ability of the algorithm [36]. The mathematical expression is shown in

$$\begin{aligned} x_1 &= \mathrm{rand}, \\ x_{i+1} &= \begin{cases} \dfrac{x_i}{a}, & \text{if } x_i < a, \\ \dfrac{x_i}{(1-a)}, & \text{if } x_i \ge a, \end{cases} \quad i = 1, 2, \cdots, D, \end{aligned} \quad (11)$$

where a is the chaos factor, and $a = 0.7$ in here. D is the dimension of population.

3.2.2. The Modified Archimedes Optimization Algorithm (MAOA). Firstly, the paper introduces tent chaotic map into the population initialization phase of AOA. Secondly, referring the concepts of social learning and individual perception in PSO to optimize the density and volume formula of AOA, the paper proposed the MAOA. The steps of MAOA are as follows:

Step 1: initialization. initialize the positions of all objects using

$$X_i = \mathrm{Tent}(N).^{*}(\mathrm{ub}_i - \mathrm{lb}_i) + \mathrm{lb}_i\, i = 1, 2, \cdots, D, \quad (12)$$

where D is the dimension of population and N is the number of the population. The ub_i and lb_i are the lower and upper bounds of the search space, respectively.

Table 1: MAOA sensitivity analysis of parameters.

Number	Parameters				Test function	
	C_3	C_4	r_1	r_2	F_5	F_{12}
1	1	1	1	0.5	2.89$E-07$	5.62$E-09$
2	1	1	1	1	2.89$E-07$	1.18$E-08$
3	1	1	1.5	0.5	2.90$E-07$	7.86$E-09$
4	1	1	1.5	1	2.90$E-07$	3.70$E-08$
5	1	1	1.8	0.5	2.89$E-07$	5.84$E-09$
6	1	1	1.8	1	2.89$E-07$	4.85$E-09$
7	1	2	1	0.5	2.89$E-07$	9.93$E-09$
8	1	2	1	1	2.89$E-07$	1.08$E-08$
9	1	2	1.5	0.5	2.90$E-07$	6.80$E-09$
10	1	2	1.5	1	2.90$E-07$	2.88$E-08$
11	1	2	1.8	0.5	2.89$E-07$	3.96$E-09$
12	1	2	1.8	1	2.89$E-07$	6.78$E-09$
13	2	1	1	0.5	1.93$E-03$	2.39$E-06$
14	2	1	1	1	5.85$E-03$	3.20$E-07$
15	2	1	1.5	0.5	5.69$E-03$	1.25$E-06$
16	2	1	1.5	1	1.07$E-02$	2.96$E-07$
17	2	1	1.8	0.5	4.04$E-03$	5.24$E-07$
18	2	1	1.8	1	6.66$E-03$	2.54$E-07$
19	2	2	1	0.5	4.34$E-03$	2.37$E-06$
20	2	2	1	1	2.18$E-03$	6.35$E-07$
21	2	2	1.5	0.5	6.78$E-03$	1.73$E-06$
22	2	2	1.5	1	2.81$E-03$	1.40$E-07$
23	2	2	1.8	0.5	2.09$E-01$	1.81$E-07$
24	2	2	1.8	1	3.10$E-03$	2.33$E-07$

Initialize density (den), volume (vol), and acceleration (acc) for the ith object using

$$\begin{aligned} \mathrm{den}_i &= \mathrm{rand}, \\ \mathrm{vol}_i &= \mathrm{rand}, \\ \mathrm{acc}_i &= \mathrm{rand}\,(N, D) \times (\mathrm{ub}_i - \mathrm{lb}_i) + \mathrm{lb}_i. \end{aligned} \tag{13}$$

Initialize the optimal position (P_i) and corresponding fitness (P_{best_i}) for the ith object using

$$\begin{aligned} P_i &= X_i, \\ P_{\mathrm{best}_i} &= \mathrm{fitness}(P_i). \end{aligned} \tag{14}$$

Then, P_{den} and P_{vol} corresponding to P and X_{best}, ac c_{best}, $\mathrm{den}_{\mathrm{best}}$, and $\mathrm{vol}_{\mathrm{best}}$ are selected in the initialized population.

Step 2: update the density and volume of population using

$$\begin{aligned} \mathrm{den}_i^{t+1} &= \mathrm{den}_i^t + r_1 \times \mathrm{rand} \times \left(P_{\mathrm{den}_i} - \mathrm{den}_i^t\right) \\ &\quad + r_2 \times \mathrm{rand} \times \left(\mathrm{den}_{\mathrm{best}} - \mathrm{den}_i^t\right), \\ \mathrm{vol}_i^{t+1} &= \mathrm{vol}_i^t + r_1 \times \mathrm{rand} \times \left(P_{\mathrm{vol}_i} - \mathrm{den}_i^t\right) \\ &\quad + r_2 \times \mathrm{rand} \times \left(\mathrm{vol}_{\mathrm{best}} - \mathrm{vol}_i^t\right), \end{aligned} \tag{15}$$

where t is the current iteration and r_1 and r_2 represent individual cognitive coefficients and social learning coefficient, respectively. If $r_1 = 0$, there is no individual cognition and the convergence speed of the algorithm is fast but fall into local optimization easily. Step 3, step 4, and step 5 are all same as AOA. After the three-step update iteration, a new population X_{new} is obtained.

Step 6: evaluate each object using fitness of objective function and remember the best solution found so far. Update the current optimal position P of each object and

TABLE 2: The test function.

Type	Function	Dim	Range	$f_{\min}$
	$F_1(x)=\sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
	$F_2(x)=\sum_{i=1}^{n} \lvert x_i\rvert+\prod_{i=1}^{n} \lvert x_i\rvert$	30	[-10,10]	0
	$F_3(x)=\sum_{i=1}^{n}\left(\sum_{j-1}^{i} x_j\right)^2$	30	[-100,100]	0
Unimodal function	$F_4(x)=\max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	30	[-100,100]	0
	$F_5(x)=\sum_{i=1}^{n-1}\left[100\times\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$	30	[-30,30]	0
	$F_6(x)=\sum_{i=1}^{n}([x_i+0.5])^2$	30	[-100,100]	0
	$F_7(x)=ix_i^4+\text{rand } m[0,1]$	30	[-128,128]	
	$F_8(x)=\sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	[-500,500]	12569.48
	$F_9(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	30	[-5.12,5.12]	0
	$F_{10}(x)=-20\exp\left(-0.2\sqrt{1/n\sum_{i=1}^{n} x_i^2}\right)-\exp\left(1/n\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	30	[-32,32]	0
	$F_{11}(x)=1/4000\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos\left(x_i/\sqrt{i}\right)+1$	30	[-600,600]	0
Multimodal function	$F_{12}(x)=(\pi/n)\left\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_{i-1})^2\left[1+\sin^2(\pi y_{i+1})\right]+(y_n-1)^2\right\}$ $y_i=1+((x_1+1)/4),$ $u=(x_i,a,k,m)=\begin{cases} k(x_i-a)^m, & x_i>a,\\ 0, & -a<x_i<a,\\ k(-x_i-a)^m, & x_i<-a \end{cases}$	30	[50,50]	0
	$F_{13}(x)=0.1\left\{\sin^2(3\pi x_i)+\sum_{i=1}^{n}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right]+(x_i-1)^2\left[1+\sin^2(2\pi x_i+1)\right]+\sum_{i=1}^{n}u(x_i,5,100,4)\right.$	30	[50,50]	0

Table 3: Parameter settings of every algorithm.

Index	Algorithm	Parameter
1	AOA	$N = 30$, $C_1 = 2$, $C_2 = 6$, $C_3 = 1$, $C_4 = 2$, $T_{\max} = 100$
2	MAOA	$N = 30$, $C_1 = 2$, $C_2 = 6$, $C_3 = 1$, $C_4 = 2$, $r_1 = 1.8$, $r_2 = 0.5$, $T_{\max} = 100$
3	BOA	$N = 30$, conversion probability $p = 0.6$, initial value of a is 0.1, $T_{\max} = 100$
4	PSO	$N = 30$, w decreases linearly in [0.2, 0.9], $c_1 = 2$, $c_2 = 2$, $T_{\max} = 100$
5	DE	$N = 30$, CR = 0.1, $T_{\max} = 100$
6	MPA	$N = 30$, FADs = 0.2, $P = 0.5$, $T_{\max} = 100$

Table 4: Test result statistics.

Function	Type	AOA	MAOA	BOA	PSO	DE	MPA
F_1	Mean	$5.639E-05$	$5.492E-09$	$9.100E-03$	$2.073E+04$	$9.588E+03$	$4.648E+01$
	Std	$1.143E-04$	$3.313E-08$	$2.000E-03$	$1.990E+03$	$1.692E+03$	$2.260E+01$
F_2	Mean	$9.569E-04$	$1.016E-05$	$3.980E-02$	$4.391E+00$	$4.021E+01$	$2.727E+00$
	Std	$1.142E-03$	$3.098E-05$	$1.420E-02$	$1.275E+00$	$3.951E+00$	$5.867E-01$
F_3	Mean	$2.454E-02$	$9.931E-08$	$8.300E-03$	$7.217E+04$	$4.328E+04$	$2.205E+03$
	Std	$4.674E-02$	$7.045E-07$	$1.900E-03$	$1.996E+04$	$6.230E+03$	$1.122E+03$
F_4	Mean	$6.146E-03$	$1.825E-05$	$1.129E-01$	$4.762E+01$	$6.458E+01$	$6.478E+01$
	Std	$7.079E-03$	$3.676E-05$	$1.540E-02$	$1.112E+00$	$4.003E+00$	$4.294E+00$
F_5	Mean	$2.892E+01$	$2.891E+01$	$2.893E+01$	$1.891E+04$	$9.356E+06$	$8.748E+02$
	Std	$1.032E-01$	$3.498E-02$	$3.510E-02$	$1.893E+04$	$3.099E+06$	$4.530E+02$
F_6	Mean	$5.511E+00$	$5.566E+00$	$5.743E+00$	$2.068E+04$	$9.574E+03$	$5.444E+01$
	Std	$4.440E-01$	$6.852E-01$	$6.237E-01$	$1.771E+03$	$1.677E+03$	$2.251E+01$
F_7	Mean	$7.405E-03$	$6.257E-03$	$1.770E-02$	$9.554E-01$	$4.545E+00$	$2.341E-02$
	Std	$6.505E-03$	$6.086E-03$	$7.300E-03$	$3.586E-01$	$1.402E+00$	$1.278E-02$
F_8	Mean	$-5.345E+03$	$-4.129E+03$	$-3.126E+03$	$-2.435E+03$	$-5.770E+03$	$-6.046E+03$
	Std	$3.835E+03$	$4.679E+02$	$3.422E+02$	$4.923E+02$	$3.927E+02$	$5.811E+02$
F_9	Mean	$2.497E+01$	$2.920E-07$	$5.846E+01$	$1.064E+02$	$2.156E+02$	$7.705E+01$
	Std	$6.370E+01$	$1.854E-06$	$8.200E+01$	$2.240E+01$	$1.713E+01$	$2.900E+01$
F_{10}	Mean	$1.637E+01$	$4.345E-06$	$9.800E-02$	$7.056E+00$	$1.574E+01$	$3.071E+00$
	Std	$7.670E+00$	$1.121E-05$	$1.690E-02$	$1.086E+00$	$6.888E-01$	$4.540E-01$
F_{11}	Mean	$2.150E-01$	$2.812E-08$	$1.238E-01$	$5.673E+02$	$8.788E+01$	$1.443E+00$
	Std	$3.083E-01$	$2.632E-07$	$3.830E-02$	$4.728E+01$	$1.376E+01$	$1.730E-01$
F_{12}	Mean	$7.449E-01$	$6.732E-01$	$7.435E-01$	$1.580E+01$	$4.280E+06$	$1.069E+00$
	Std	$1.868E-01$	$1.843E-01$	$1.464E-01$	$1.449E+01$	$3.028E+06$	$4.527E-01$
F_{13}	Mean	$2.958E+00$	$2.937E+00$	$3.086E+00$	$9.261E+03$	$2.033E+07$	$4.866E+00$
	Std	$1.048E-01$	$1.079E-01$	$1.896E-01$	$1.508E+04$	$8.453E+06$	$1.489E+00$

Table 5: Comparison of execution time.

Function	Time of AOA		Time of MAOA	
	Total	Average	Total	Average
F_1	0.385	$1.283E-02$	0.503	$1.677E-02$
F_2	0.458	$1.527E-02$	0.551	$1.837E-02$
F_3	0.843	$2.810E-02$	0.961	$3.203E-02$
F_4	0.423	$1.410E-02$	0.525	$1.750E-02$
F_5	0.478	$1.593E-02$	0.592	$1.973E-02$
F_6	0.413	$1.377E-02$	0.534	$1.780E-02$
F_7	0.668	$2.227E-02$	0.775	$2.583E-02$
F_8	0.498	$1.660E-02$	0.612	$2.040E-02$
F_9	0.463	$1.543E-02$	0.580	$1.933E-02$
F_{10}	0.468	$1.560E-02$	0.597	$1.990E-02$
F_{11}	0.515	$1.717E-02$	0.632	$2.107E-02$
F_{12}	1.091	$3.637E-02$	1.217	$4.057E-02$
F_{13}	1.112	$3.707E-02$	1.234	$4.113E-02$

the corresponding $P_{\text{best}}, P_{\text{den}}, \text{and}\, P_{\text{vol}}$. Then, update the global optimal individual X_{best} and corresponding acc_{best}, den_{best}, and vol_{best}.

3.3. Algorithm Simulation Analysis. In this section, we compared the performance of MAOA with five algorithms, including AOA, BOA, PSO, DE, and MPA by 13 test functions. The global convergence ability and convergence speed of the proposed MAOA are analyzed to verify the optimization ability.

Simulation platform: Windows 10 system using Intel (R) Core (TM) i5-10210U CPU @16 G RAM, and MATLAB 2018b.

3.3.1. Parameter Setting of MAOA. In the improved algorithm, the parameters including $C_1, C_2, C_3, C_4, r_1, \text{and}\, r_2$ need to be set. According to the analysis of AOA algorithm in reference [23], we choose $C_1 = 2, C_2 = 6$. The sensitivity of the other four parameters of MAOA is analyzed by unimodal function F_5 and multimodal test function F_{12}.

Setting the parameter $N = 30$ and $\max - \text{iterations} = 200$, the test results are shown in Table 1. It is obvious from the test results in Table 1 that the performance of MAOA is the best while $C_3 = 1,\ C_4 = 2,\ r_1 = 1.8, \text{and}\, r_1 = 0.5$.

3.3.2. MAOA Convergence Analysis. We compared the proposed algorithm with AOA, BOA, PSO, DE, and MPA through 13 test functions. The test functions are shown in Table 2, including seven unimodal functions, $F_1 - F_7$, and six multimodal functions, $F_8 - F_{13}$. The dimensions of the test function are all set to 30. To avoid the error resulted by accidental factors, the average value of the results of 100 runs of each algorithm is used to measure the optimization performance of each algorithm. The parameters of each algorithm are shown in Table 3, and the test results are shown in Table 4.

In Table 4, the 'Mean' represents the average value of the optimization results and the 'Std' represents the standard deviation of the optimization results, and the blackened one is the minimum value. Compared with other algorithms, MAOA outperformed other metaheuristic algorithms for 11 functions including $F_1, F_2, F_3, F_4, F_5, F_7, F_9, F_{10}, F_{11}, F_{12}, \text{and}\, F_{13}$. In the process of 100 tests, the standard deviation of the test result of MAOA algorithm is relatively small. This indicates that the stably global convergence ability of the algorithm, so the improvement of AOA is effective and MAOA has better global convergence ability than others.

3.3.3. Complexity Analysis of Algorithm

(1) Complexity analysis of AOA

The time complexity of AOA is given as follows: Assuming that the population size is n, the dimension of search space is d, and the maximum iteration is T. The population density, volume, acceleration, and initialization complexity are all $O(nd)$, the fitness value of calculation complexity is $O(nd)$. The exploration and exploitation phases update complexity are $O(T(1 + 2n + 2n \log n)$. To sum up, the time complexity of the whole algorithm is expressed as

$$O(\text{AOA}) = 5 \times O(nd) + O(T(1 + 2n + 2n \log n)). \quad (16)$$

(2) Complexity analysis of MAOA

The time complexity of MAOA is given as follows: the population, density, volume, P, and acceleration initialization complexity are all $O(nd)$, and the fitness value of calculation complexity is $O(nd)$. The exploration and exploitation phases

```
1:  Procedure MAOADV-Hop;
2:  Initialization: total number of nodes $N$, percentage $p$ of anchor nodes, communication radius $R$;
3:  Network deployment nodes to generate simulated
network topology;
4:      Calculate the hop-count value $h_{i,j}$ according to the shortest path algorithm;
5:  for $k = 1$ to $N$
6:      for $i = 1$ to $N$
7:          for $j = 1$ to $N$
8:              if short_path($i, k$)+short_path($k, j$)<short_path($i, j$);
9:                  short_path($i, j$)=short_path($i, k$)+short_path($k, j$);
10:                     end
11:                 end
12:             end
13: end
14:     Calculate the average distance Hop-size of each hop using Equation (18);
15:     Calculate the estimated distance from the anchor node to the unknown node using Equation (19);
16:     Initialize the parameters at the population level of MAOA algorithm using (12)–(14);
17: for $t = 1$: Max_iter
18:     Calculate TF using Equation (3) and calculate density decline coefficient $d$ using Equation (4);
19:     for $i = 1 : N$
20:         Update $\text{den}_i^{t+1}$ and $\text{vol}_i^{t+1}$ using Equation (15);
21:         if TF < 0.5
22:                 Update $\text{acc}_i^{t+1}$ using Equation (5) and calculate $X_i^{t+1}$ using Equation (8);
23:         else
24:             Update $\text{acc}_i^{t+1}$ using Equation (6) and calculate $X_i^{t+1}$ using Equation (9);
25:         end
26: Apply boundary constraints to $X$ and calculate the fitness of $X$;
27: The optimal position of each object is selected and assigned to $P$, and the corresponding fitness is assigned to$P_{\text{best}}$;
28:     Select the best fitness minimum value, the corresponding position $X_{\text{best}}$;
29: end
30:     The best individual is the location of the unknown node.
31: end
```

Algorithm 1: The pseudocode of MAOADV-Hop.

update complexity are $O(T(3 + 2n + 2n \log n)$. To sum up, the time complexity of the whole algorithm is expressed as

$$O(\text{MAOA}) = 6 \times O(nd) + O(T(3 + 2n + 2n \log n)). \quad (17)$$

From Equations (16) and (17), compared with the time complexity of AOA, the time complexity of MAOA only increases $\text{O}(nd) + \text{O}(\text{T}(2))$.

(3) Complexity contrast between AOA and MAOA

AOA and MAOA were optimized over 13 test functions for 30 times, and the total time and the average time per run were compared, and the results are shown in Table 5.

In Table 5, the unit of all data is seconds. The results are shown as follows: the total time of MAOA is longer than that of AOA about 0.115 seconds, and the average time per run of MAOA is longer than that of AOA about $3.840E - 03$ seconds. Therefore, the time complexity of MAOA increased is not obvious, compared with that of AOA.

4. MAOADV-Hop Algorithm

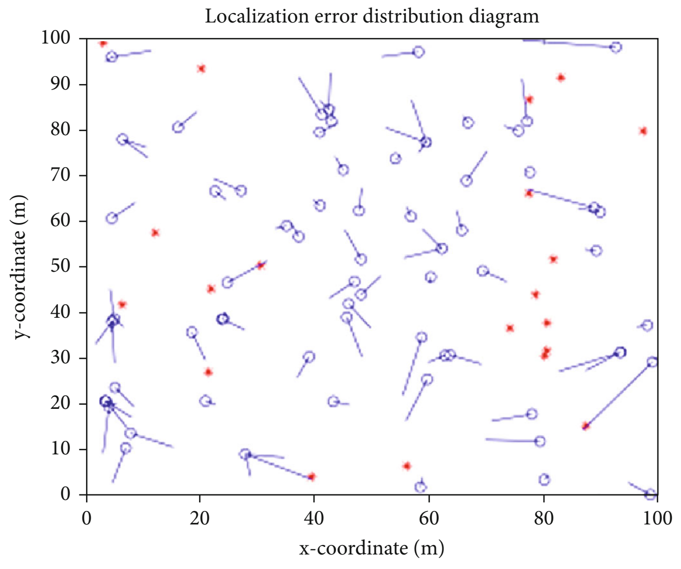
4.1. The Node Localization Algorithm. In this section, an improved localization algorithm, MAOADV-Hop, is proposed by introducing MAOA into DV-Hop. The step of the algorithm is shown as follows.

Step 1. Flooding. All anchor nodes broadcast data packets from their locations to their neighbors, and those neighbors of the receiving packet send the new packet to the other neighbors. Finally, we can gain the minimum hop count between every anchor node and every unknown node.

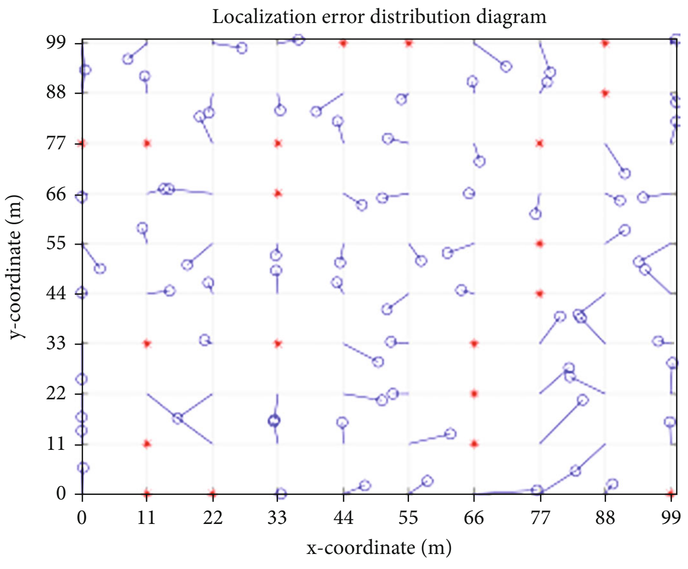
Step 2. Distance estimation between nodes. Equation (18) is used to calculate the average distance per hop:

$$\text{HopSize}_i = \frac{\sum_{i \neq j} d_{i,j}}{\sum_{i \neq j} h_{i,j}}, \quad (18)$$

where $d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, (x_i, y_i) and (x_j, y_j) are the location of anchor node i and j, respectively. $h_{i,j}$ denotes the value of minimum hop-count between anchor

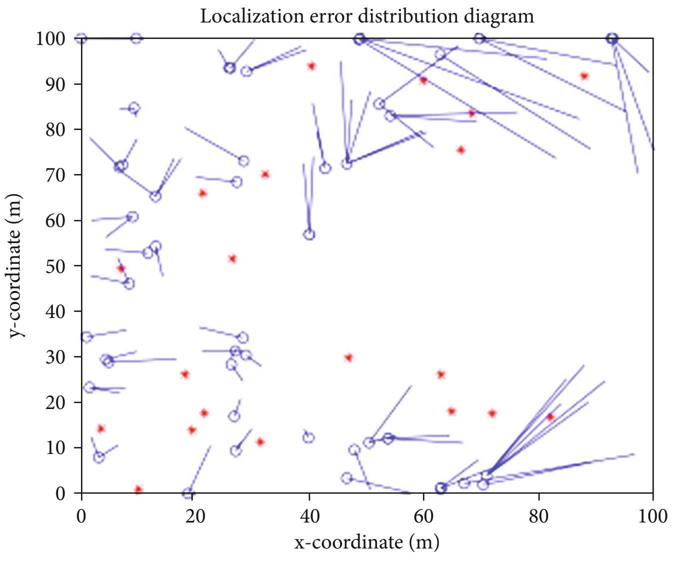


(a)

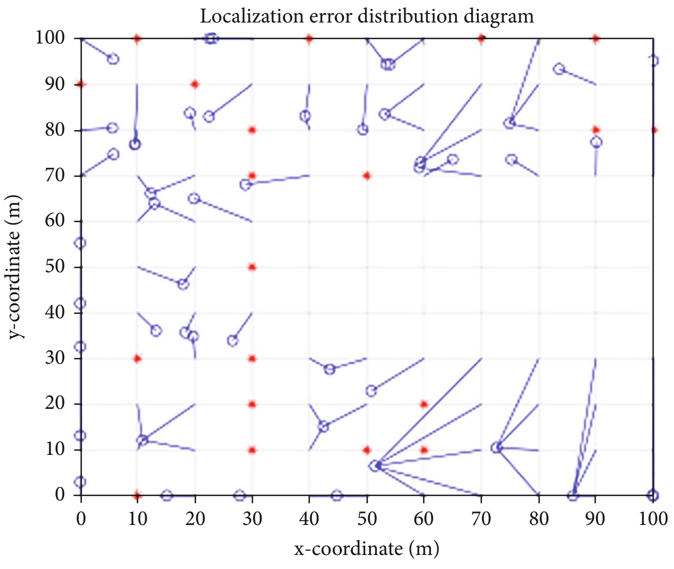


(b)

Figure 1: Continued.



(c)



(d)

FIGURE 1: Localization error diagram for (a) random deployment of square area, (b) uniform deployment of square area, (c) random deployment of C-shaped area, and (d) uniform deployment of C-shaped area.

nodes i and j. Then, Equation (19) is used to calculate the estimated distance between the anchor node i and the unknown node u.

$$d_{u,i} = \text{HopSize}_i \times h_{u,i}, \tag{19}$$

where $h_{u,i}$ denotes the value of minimum hop count between unknown node u and the anchor node i.

Step 3. It uses MAOA to search the optimal solution of fitness $f(x_u) = \sum_{i=1}^{m} \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2} - d_{u,i}$, where (x_u, y_u) represents the estimated location of unknown node u and (x_i, y_i) represents the actual location of unknown node u. m is the number of the anchor node. The estimated position corresponding to the optimal solution is the localization of the unknown node.

The pseudocode of MAOADV-Hop is shown in Algorithm 1.

4.2. Experimental Results. In this section, the simulation is carried out in the square area and C-shaped area of 100 m × 100 m deploying sensor nodes by random or uniform. The C-shaped area is formed by digging out a rectangular area of 30 m × 70 m in the square area. MAOADV-Hop is

Table 6: Parameter settings of DE_DV-Hop.

Parameters	Value
NP	30
Max-iteration	200
F_0	0.5
CR	0.2

Table 7: Parameter settings of BOA_DV-Hop.

Parameters	Value
NP	30
Max-iteration	200
Probability(p)	0.6
Initial value of a, c	0.1, 0.01

Table 8: Parameter settings of MAOA_DV-Hop.

Parameters	Value
NP	30
Max-iteration	200
$C1, C2, C3, C4$	2, 6, 1, 2
$r1, r2$	1.8, 0.5

used to estimate the location of the unknown node. The simulation results are shown in Figure 1.

In Figure 1, the red '*' represents the anchor node. The blue 'O' represents the estimated location of the unknown nodes. The blue line connects the estimated location and the actual location of the same unknown node and its length is the localization error size. Only considered 2D coordinate planes, all experimental results were run independently for 100 times to calculate the average value. It uses the average localization error (ALE) to measure the localization accuracy.

$$\mathrm{ALE} = \frac{\sqrt{(x_u^{\mathrm{ext}} - x_u^{\mathrm{act}})^2 + (y_u^{\mathrm{ext}} - y_u^{\mathrm{act}})^2}}{n \times R}, \qquad (20)$$

where $(x_u^{\mathrm{ext}}, y_u^{\mathrm{ext}})$ and $(x_u^{\mathrm{act}}, y_u^{\mathrm{act}})$ represent the estimate location and the actual location of the unknown node u, n is the number of unknown nodes, and R is the communication radius between nodes.

4.3. Localization Algorithm Analysis

4.3.1. Compared with Different Algorithms. We compared the localization error with DV-Hop, DE_DV-Hop, BOA_DV-Hop, and MAOADV-Hop in order to measure the performance of the proposed localization algorithm. The parameter settings of MAOADV-Hop, DE_DV-Hop, and BOA_DV-Hop are shown in Tables 6–8.

Experimental analysis is carried out under four different simulation conditions: square random deployment, square uniform deployment, C-shaped random steps, and C-shaped uniform deployment to prove the localization accuracy and convergence speed of the proposed algorithm. The experimental results are shown in Figures 2 and 3.

From Figures 2 and 3, the average localization error of MAOADV-Hop is 35% to 71% lower than DV-Hop and is 3% to 7% lower than BOA_DV-Hop. The convergence rate of MAOADV-Hop is faster than both DE-DV-Hop and BOA_DV-Hop. The localization error curve of MAOADV-Hop in the four scenarios is relatively smooth, tends to be horizontal after just several iterations, and has a good capability of global optimization. DE_DV-Hop fell into the local optimal value in a short period of time after 25 iterations. Therefore, the localization performance of MAOADV-Hop is better than the other three localization algorithms.

4.3.2. Effect of Node Density, Anchor Rate, and Communication Radius on ALE

(1) The effect of node density on ALE

In this section, the experimental area is a 100 m × 100 m square area and the percentage of anchor nodes is 20%. The number of square topology deployment nodes is 80 to 200 and the communication radius is 20 m. Experimental results are shown in Figure 4.

From Figure 4, with the increasing in the number of deployed nodes, the ALES of four localization algorithms are all decreased. Because of the increase in the number of nodes, the connectivity of the network is enhanced, and the estimation distance error between unknown nodes and anchor nodes is reduced. Therefore, the localization accuracy is stronger.

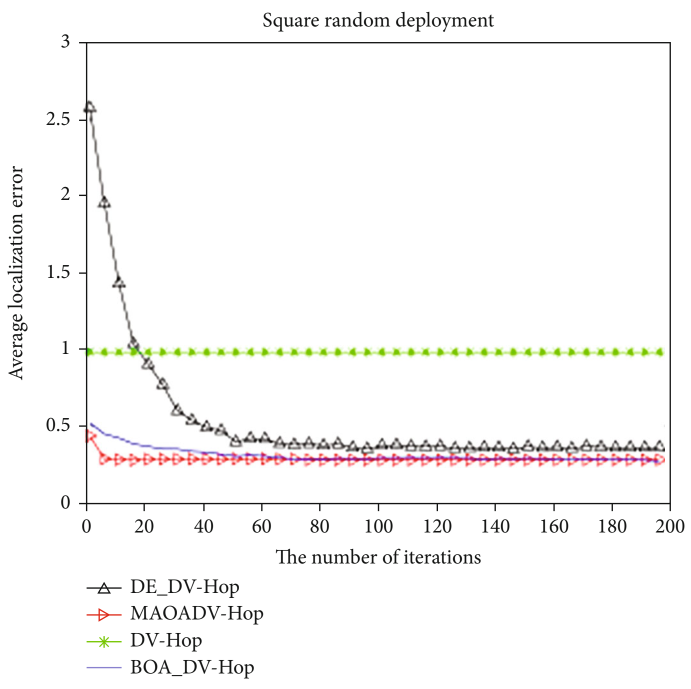
(2) The effect of anchor rate on ALE

In this section, the experimental area is a 100 m × 100 m square area and the percentage of anchor nodes is 15% to 35%. The number of square topology deployment nodes is 100, and the communication radius is 20 m. Experimental results are shown in Figure 5.

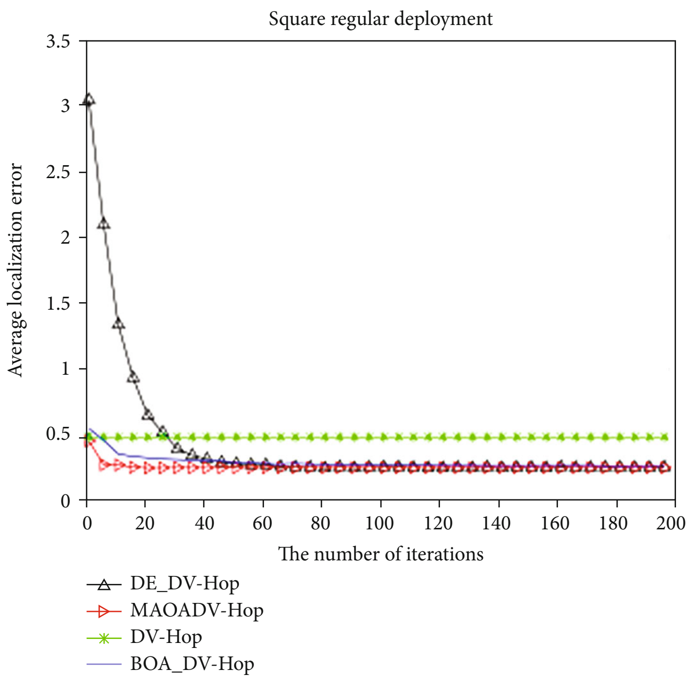
From Figure 5, with the increasing ratio of anchor nodes, the ALE of four localization algorithms is all decreased. Because of the increasing ratio of anchor nodes, the accuracy of the average distance per hop of the anchor nodes is higher. The estimated distances between the anchor nodes and the unknown nodes are closer to the real distance. And the increase of the number of anchors provides more conditions for the metaheuristic algorithm, so it can estimate the location of unknown nodes more accurately.

(3) The effect of communication radius on ALE

In this section, the experimental area is a 100 m × 100 m square area and the percentage of anchor nodes is 20%. The number of the square topology deployment nodes is 100, and the communication radius is 15 m to 45 m. Experimental results are shown in Figure 6.

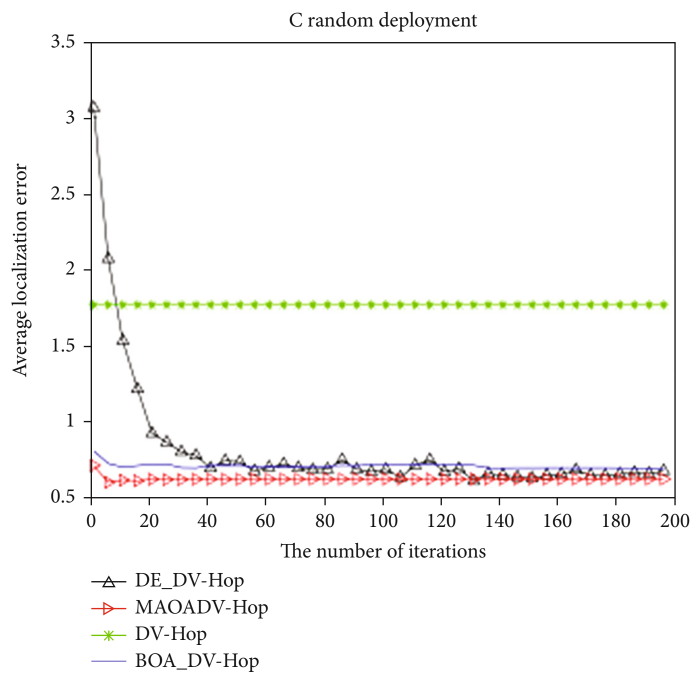


(a)

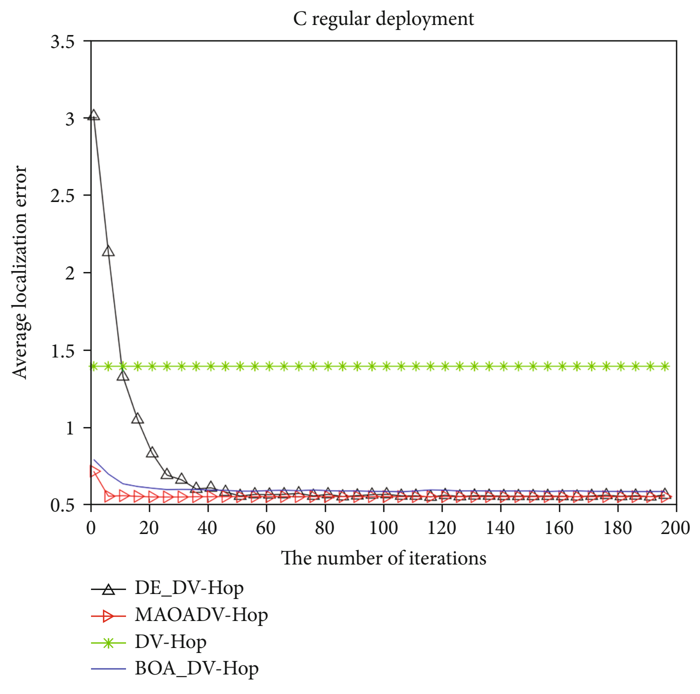


(b)

Figure 2: Continued.



(c)



(d)

Figure 2: (a) Random deployment in square area. (b) Uniform deployment in square area. (c) Random deployment in C-shaped area. (d) Uniform deployment in C-shaped area.

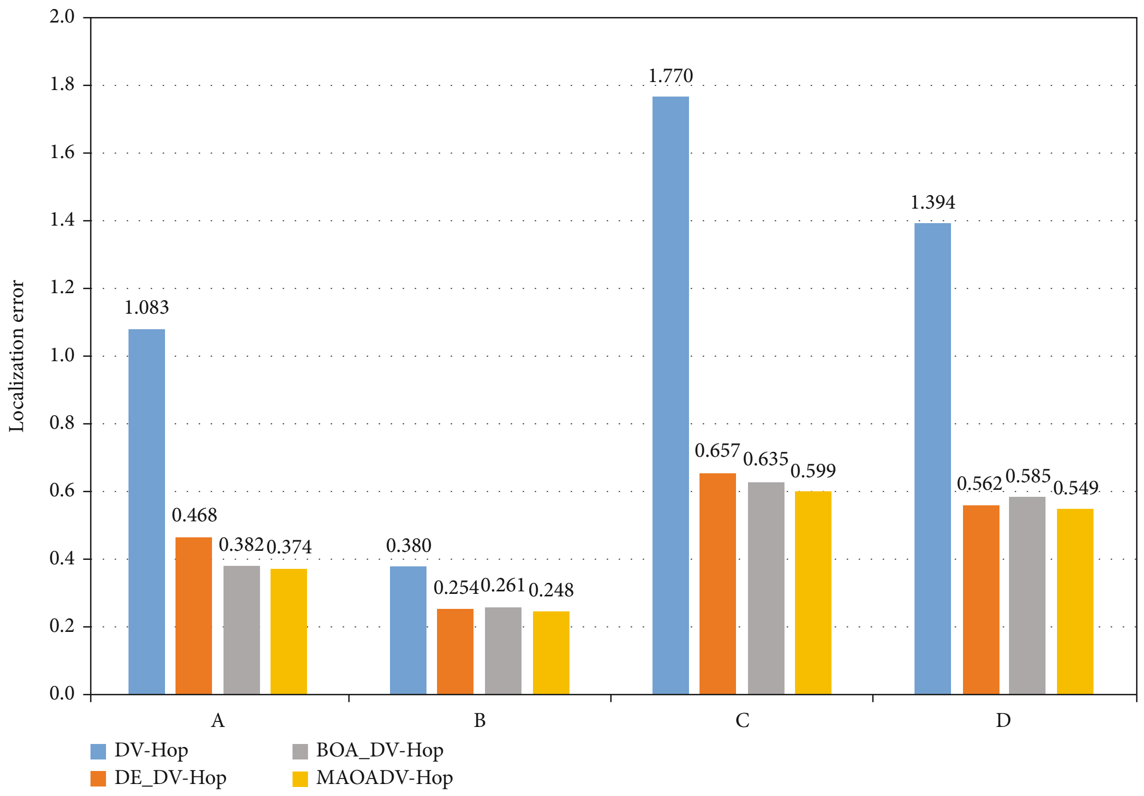


Figure 3: The average value of ALE for the four topologies and algorithms.

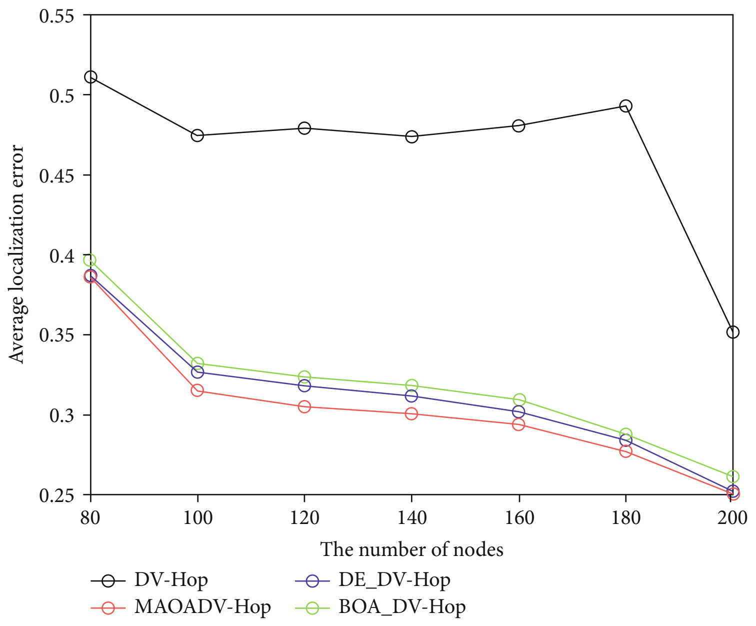


Figure 4: The effect of node density on ALE.

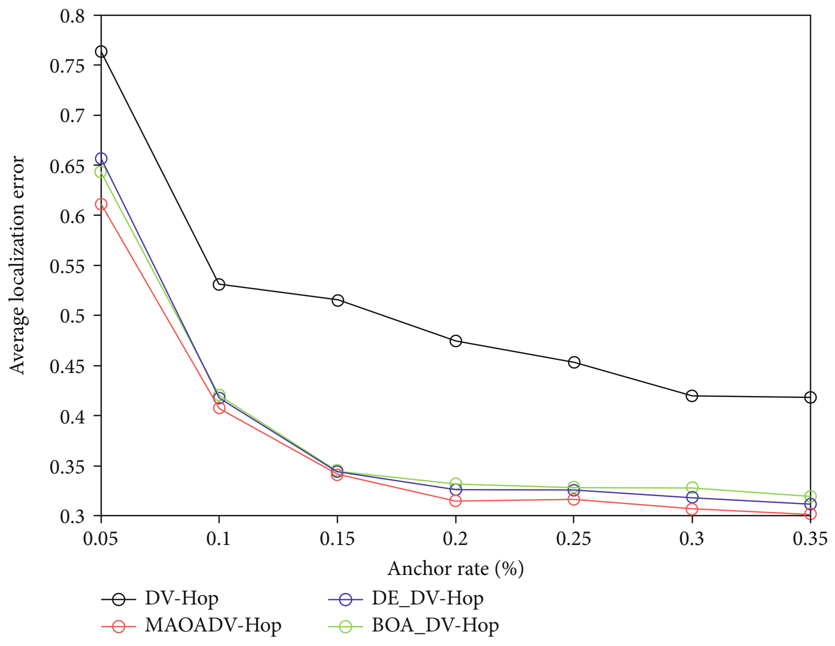


Figure 5: The effect of anchor rate on ALE.

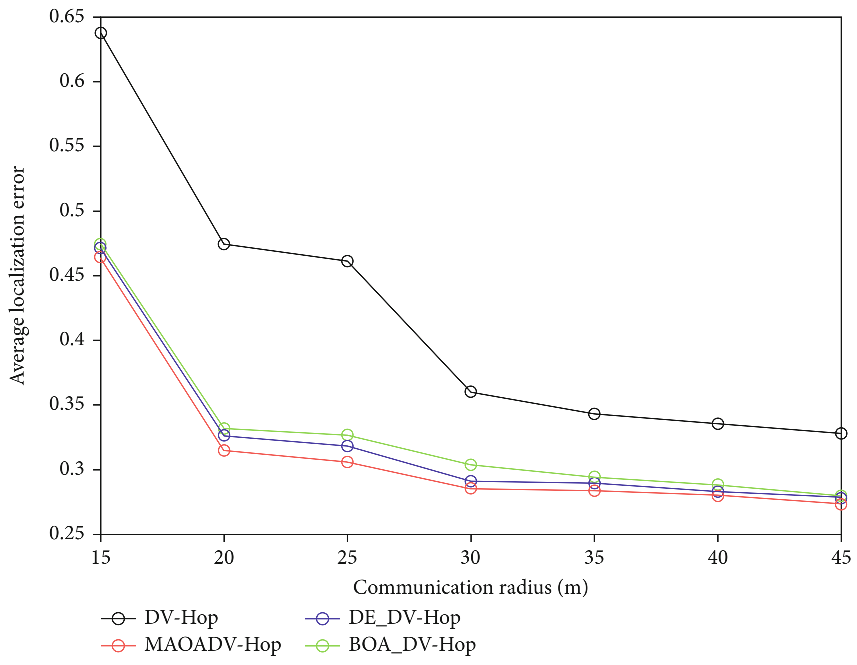


Figure 6: The effect of communication radius on ALE.

In Figure 6, with the increase of communication radius, the ALE of the four localization algorithms decreased. More nodes can communicate in a single-hop way, which reduces the hop value between unknown nodes and anchor points because of the bigger communication radius. The distance between the unknown node and the anchor point is estimated more accurately. And the estimated location of unknown nodes is more close real location. In addition, the localization

Table 9: The parameter settings of WSNs.

Parameter	Value
Manner of deployment	Random deployment
The number of nodes	100
Anchor rate	0.2
Communication radius	20

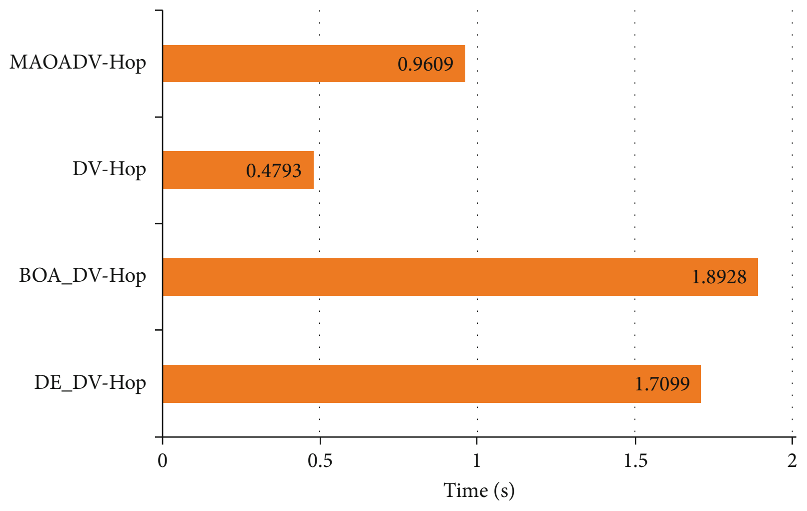


Figure 7: The CPU time required of different algorithms.

error of MAOADV-Hop is the smallest in the four algorithms although communication radius is different.

4.3.3. Timing of Localization. In this section, we analyze the CPU time required of the four localization algorithms. The WSN parameter settings are shown in Table 9. DE_DV-Hop, BOA_DV-Hop, DV-Hop, and MAOADV-Hop are used to locate the unknown nodes. The CPU time required of each algorithm is shown in Figure 7.

From Figure 7, it is obvious that the time by DV-Hop is the shortest with 0.4793 seconds, followed by the time used by MAOADV-Hop, which is 0.9609 seconds. The time required by DE_DV-Hop and BOA_DV-Hop is longer, which is about 4 times of DV-Hop and twice of MAOADV-Hop. Although the time of DV-Hop is shorter than MAOADV-Hop, the localization accuracy of MAOADV-Hop is much higher than that of DV-Hop. Therefore, MAOADV-Hop is the best way to locate unknown nodes in the four localization algorithms.

5. Conclusions and Future Work

5.1. Conclusions

(1) In this paper, a modified Archimedes optimization algorithm (MAOA) is proposed, and seven unimodal functions and six multimodal functions are used to test the convergence speed and the optimization ability of the algorithm. The test results show that the convergence and optimization ability of the MAOA is better than AOA, DE, BOA, PSO, and MPA

(2) The paper proposes a localization algorithm based on MAOA and DV-Hop, called MAOADV-Hop. The simulation results show that the localization effect of MAOADV-Hop is better than that of DE_DV-Hop, BOA_DV-Hop, and DV-Hop, such as better localization accuracy and faster speed

5.2. Future Work. This paper only discusses the application of node localization based on the two-dimensional plane in WSNs. These factors associated with WSNs that need futuristic attention are listed as follows.

(1) More complex spaces: the future work will focus on the study of localization in the three-dimensional WSNs

(2) Energy consumption: the future work should integrate the routing algorithm with localization in order to reduce the energy consumption, which is helpful of extending the lifetime of WSNs

Data Availability

The simulation program used to support the findings of this study have been deposited in the GitHub repository (https://chengmang1.github.io/data2/).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are grateful for the support provided by the Guizhou Provincial Key Laboratory of Internet+Intelligent Manufacturing, Guiyang 550025, China. This work is supported by the National Natural Science Foundation of China (under Grant Nos. 61640014 and 61963009), Innovation group of Guizhou Education Department (No. Qianjiaohe KY[2021]012), Industrial Project of Guizhou province (Grant No. Qiankehe Zhicheng [2022]Yiban017, [2019] 2152), Science and Technology Fund of Guizhou Province (No. Qiankehejichu [2020]1Y266), Plan of Guizhou province Qiankehe-Pingtai (No. (2017) 5788), and the platform of IOT personnel from Guiyang high technology industry development zone under Grant 2015, postgraduate case library (under Grant No. KCALK201708).

References

[1] G. Bhatti, "Machine learning based localization in large-scale wireless sensor networks," *Sensors*, vol. 18, no. 12, pp. 4179–4199, 2018.

[2] Z. Li and A. Zhong, "Resource allocation in wireless powered virtualized sensor networks," *IEEE Access*, vol. 8, pp. 40327–40336, 2020.

[3] W. Liang, M. Tang, J. Long, X. Peng, J. Xu, and K. C. Li, "A secure FaBric blockchain-based data transmission technique for industrial Internet-of-Things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3582–3592, 2019.

[4] M. S. Perumal, B. Manimozhi, H. Dandamudi, V. B. Durairaj, and A. Jawaharlalnehru, "Ultra-reliable low latency communication technique for agriculture wireless sensor networks," *Arabian Journal of Geosciences*, vol. 14, no. 13, pp. 1246–1255, 2021.

[5] S. Singh, A. Malik, and P. K. Singh, "A threshold-based energy efficient military surveillance system using heterogeneous wireless sensor networks," *Soft Computing*, vol. 26, no. 7, 2021.

[6] V. Kavitha and S. Mohanraj, "Green engineering principles for global water quality monitoring using IoT," *International Journal of Environment and Sustainable Development*, vol. 18, no. 1, pp. 120–129, 2019.

[7] A. Hilmani, A. Maizate, and L. Hassouni, "Automated real-time intelligent traffic control system for smart cities using wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2020, 28 pages, 2020.

[8] M. Cui, D. Han, and J. Wang, "An efficient and safe road condition monitoring authentication scheme based on fog computing," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9076–9084, 2019.

[9] W. Liang, K. Li, J. Long, X. Kui, and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063–2071, 2020.

[10] X. Su, Z. Cai, X. Jia, L. Guo, and Z. Ding, "A self-adaptive approach for mobile wireless sensors to achieve energy efficient information transmission," *IEEE Access*, vol. 8, pp. 86296–86304, 2020.

[11] M. Li, F. Jiang, and C. Pei, "Review on positioning technology of wireless sensor networks," *Wireless Personal Communications*, vol. 115, no. 3, pp. 2023–2046, 2020.

[12] J. Luo, Y. Yang, Z. Wang, and Y. Chen, "Localization algorithm for underwater sensor network: a review," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13126–13144, 2021.

[13] S. Kumar, N. Batra, and S. Kumar, "Range-independent localization for GPS dead zone in MWSN," *Wireless Networks*, vol. 27, no. 7, pp. 4807–4823, 2021.

[14] J. Kumari, P. Kumar, and S. K. Singh, "Localization in three-dimensional wireless sensor networks: a survey," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 5040–5083, 2019.

[15] P. Xie, K. You, S. Song, and C. Wu, "Distributed range-free localization via hierarchical nonconvex constrained optimization," *Signal Processing*, vol. 164, no. 11, pp. 136–145, 2019.

[16] I. Dimitriou and N. Pappas, "Performance analysis of a cooperative wireless network with adaptive relays," *Ad Hoc Networks*, vol. 87, pp. 157–173, 2019.

[17] A. Kargar-Barzi and A. Mahani, "H–V scan and diagonal trajectory: accurate and low power localization algorithms in WSNs," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 7, pp. 2871–2882, 2020.

[18] G. Qiang, Z. Guohui, and L. Wanchen, "TDOA/FDOA localization based on chaotic sparrow search algorithm," *Journal of Jilin University (Engineering Edition)*, vol. 23, no. 6, pp. 1–9, 2021.

[19] S. Kang, T. Kim, and W. Chung, "Multi-target localization based on unidentified multiple RSS/AOA measurements in wireless sensor networks," *Sensors*, vol. 21, no. 13, pp. 4455–4470, 2021.

[20] Z. Huansheng, C. Bingde, and F. Tao, "Node localization algorithm based on hop vector in mobile WSNs," *Firepower and command and control*, vol. 46, no. 5, pp. 142–146, 2021.

[21] E. Shakshuki, A. A. Elkhail, I. Nemer, M. Adam, and T. Sheltami, "Comparative study on range free localization algorithms," *Procedia Computer Science*, vol. 151, pp. 501–510, 2019.

[22] I. Nemer, T. Sheltami, E. Shakshuki, A. A. Elkhail, and M. Adam, "Performance evaluation of range-free localization algorithms for wireless sensor networks," *Personal and Ubiquitous Computing*, vol. 25, no. 1, pp. 177–203, 2021.

[23] R. T. Moorthi and R. Thiagarajan, "Energy consumption and network connectivity based on Novel-LEACH-POS protocol networks," *Computer Communications*, vol. 149, no. 1, pp. 90–98, 2020.

[24] A. Ouyang, Y. Lu, Y. Liu, M. Wu, and X. Peng, "An improved adaptive genetic algorithm based on DV-Hop for locating nodes in wireless sensor networks," *Neurocomputing*, vol. 458, no. 1, pp. 500–510, 2021.

[25] J. Cota-Ruiz, R. Gonzalez-Landaeta, J. D. Diaz-Roman, B. Mederos-Madrazo, and E. Sifuentes, "A weighted and distributed algorithm for multi-hop localization," *International Journal of Distributed Sensor Networks*, vol. 15, no. 7, Article ID 155014771986041, 2019.

[26] D. Zhang, X. Zhang, and F. Xie, "Research on location algorithm based on beacon filtering combining DV-hop and multidimensional support vector regression," *Sensors*, vol. 21, no. 16, pp. 5335–5346, 2021.

[27] S. Messous, H. Liouane, and N. Liouane, "Improvement of DV-Hop localization algorithm for randomly deployed wireless sensor networks," *Telecommunication Systems*, vol. 73, no. 1, pp. 75–86, 2020.

[28] Q. Shi, Q. Xu, and J. Zhang, "An improved DV-Hop scheme based on path matching and particle swarm optimization algorithm," *Wireless Personal Communications*, vol. 104, no. 4, pp. 1301–1320, 2019.

[29] D. Han, Y. Yu, K. Li, and R. F. de Mello, "Enhancing the sensor node localization algorithm based on improved DV-hop and DE algorithms in wireless sensor networks," *Sensors*, vol. 20, no. 2, pp. 343–367, 2020.

[30] Y. Huang and L. Zhang, "Weighted DV-Hop localization algorithm for wireless sensor network based on differential evolution algorithm," in *2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT)*, Kahului, HI, USA, 2019.

[31] Y. Lei, G. De, and L. Fei, "Improved sparrow search algorithm based DV-Hop localization in WSN," in *2020 Chinese Automation Congress (CAC)*, Shanghai, China, 2020.

[32] X. Huang, D. Han, M. Cui, G. Lin, and X. Yin, "Three-dimensional localization algorithm based on improved A∗ and DV-hop algorithms in wireless sensor network," *Sensors*, vol. 21, no. 2, pp. 448–470, 2021.

[33] D. Han, J. Wang, C. Tang, T. H. Weng, K. C. Li, and C. Dobre, "A multi-objective distance vector-hop localization algorithm based on differential evolution quantum particle swarm optimization," *International Journal of Communication Systems*, vol. 34, no. 14, pp. 1–17, 2021.

[34] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, no. 3, pp. 1531–1551, 2021.

[35] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing*, vol. 23, no. 3, pp. 715–734, 2019.

[36] Y. Li, M. Han, and Q. Guo, "Modified whale optimization algorithm based on tent chaotic mapping and its application in structural optimization," *KSCE Journal of Civil Engineering*, vol. 24, no. 12, pp. 3703–3713, 2020.