ICIC Express Letters ICIC International ©2019 ISSN 1881-803X
Volume **13**, Number **12**, December **2019** pp. 1169–1176

AN IMPROVED TDOA ALGORITHM BASED ON CS-BPNN

Xing Huang[1,3], Chong Shen[1,3,*] and Kun Zhang[1,2,3]

[1]State Key Laboratory of Marine Resources Utilization in South China Sea
[3]College of Information Science and Technology
Hainan University
No. 58, Renmin Avenue, Haikou 570228, P. R. China
*Corresponding author: sc_hainu@163.com; kunzhang@hainu.edu.cn

[2]Education Center of MTA
Hainan Tropical Ocean University
No. 1, Yucai Road, Sanya 572022, P. R. China

Received June 2019; accepted September 2019

Abstract. *Aiming at the problem of low positioning accuracy caused by NLOS (Non-Line-of-Sight) error, this paper proposes a TDOA (Time Difference of Arrival) positioning algorithm based on CS (Cuckoo Search) and BPNN (Back Propagation Neural Network). In order to solve the TDOA non-linear equations estimation problem and NLOS denoising problem, the cuckoo search algorithm is used to estimate the TDOA non-linear equations, and the estimated results are used as the input of the trained BP neural network for correction. The output of the BP neural network is the final location result. The simulation results show that the accuracy of this algorithm is significantly improved compared with the traditional algorithm.*

Keywords: Wireless location, Cuckoo search algorithms, BP neural network, Time Difference of Arrival (TDOA)

1. **Introduction.**

With the rapid development of wireless devices, wireless positioning technology has been widely used. The commonly used wireless positioning technologies are Angle of Arrival (AOA), Time of Arrival (TOA) and Time Difference of Arrival (T-DOA) [1]. Chan algorithm and Taylor algorithm are the main algorithms for TDOA implementation. The disadvantage of these algorithms is that although Chan algorithm is fast in calculation, its positioning performance is significantly reduced in NLOS (Non-Line-of-Sight) environment; Taylor algorithm needs an initial value close to the actual value. Because the channel environment in wireless communication is complex and vulnerable to NLOS error, it usually produces large error. In recent years, intelligent algorithms have been gradually applied to positioning algorithms, for example, a PSO-NEWTON algorithm proposed in [2] and a GA-LS algorithm proposed in [3]. However, both methods require higher accuracy of initial values.

This paper proposes a CS-BP algorithm, which combines CS (Cuckoo Search) algorithm with BPNN (Back Propagation Neural Network). We will start from the estimation problem of TDOA non-linear equations, by using the cuckoo search algorithm to estimate relatively stable and accurate results and use it as the input of BPNN, and then combined with powerful function approximation ability of BPNN to eliminate the error caused by NLOS obstacles, so as to obtain relatively accurate final results. Chan algorithm and Taylor algorithm are commonly used to solve TDOA non-linear problems, so in this paper we will compare the improved algorithm with them. This algorithm does not need very accurate initial values, but only needs to ensure the stability of data for training BP

DOI: 10.24507/icicel.13.12.1169

neural network. The algorithm effectively reduces the influence of NLOS error, improves the positioning accuracy, and has good engineering significance.

2. TDOA Location Model.

Suppose that we have M base stations, $(X_i,\ Y_i)$ is the position coordinate of base station BS_i and $(x,\ y)$ is the position coordinate of mobile station MS. The distance between mobile station MS and base station BS_i is R_i. We take the first base station BS_1 as the reference base station, so R_{i1}^0 represents the range difference between the base station BS_i and the reference base station BS_1. There are the following formulas:

$$R_{i1} = c\tau_{i1} = R_{i1}^0 + cn_{i1} = R_i - R_1 + cn_{i1}, \quad i = 2, \ldots, M \tag{1}$$

where R_{i1}^0 and R_1 represent actual values, c is the transmission speed of the radio wave, τ_{i1} is the measured value of TDOA, and n_{i1} is the noise error generated during the measurement. In order to facilitate our research, we assume that n_{i1} is a Gaussian white noise with independent and identically distributed variance σ^2.

$$R_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2} \tag{2}$$

so

$$R_{i1} = \sqrt{(X_i - x)^2 + (Y_i - y)^2} - \sqrt{(X_1 - x)^2 + (Y_1 - y)^2} + cn_{i1}, \quad i = 2, \ldots, M \tag{3}$$

We can calculate the position of the mobile station by solving the non-linear equations composed of Equation (3). Here, we remember

$$\begin{aligned}
\overrightarrow{\Delta R} &= [R_{21}, R_{31}, \ldots, R_{M1}]^T_{(M-1)\times 1} \\
\overrightarrow{R_1} &= [R_1, R_1, \ldots, R_1]^T_{(M-1)\times 1} \\
\overrightarrow{R} &= [R_2, R_3, \ldots, R_M]^T_{(M-1)\times 1} \\
\overrightarrow{n} &= [n_{21}, n_{31}, \ldots, n_{M1}]^T_{(M-1)\times 1}
\end{aligned}$$

so

$$\begin{aligned}
\overrightarrow{\Delta R} &= \overrightarrow{R} - \overrightarrow{R_1} + \overrightarrow{cn} \\
&= \begin{bmatrix}
\sqrt{(X_2 - x)^2 + (Y_2 - y)^2} & - & \sqrt{(X_1 - x)^2 + (Y_1 - y)^2} \\
\sqrt{(X_3 - x)^2 + (Y_3 - y)^2} & - & \sqrt{(X_1 - x)^2 + (Y_1 - y)^2} \\
 & \vdots & \\
\sqrt{(X_M - x)^2 + (Y_M - y)^2} & - & \sqrt{(X_1 - x)^2 + (Y_1 - y)^2}
\end{bmatrix} + \overrightarrow{cn}
\end{aligned} \tag{4}$$

In this paper, we consider the case of $M > 3$ [2], using the maximum likelihood estimation to calculate the coordinate value of the point to be located. In Equation (4), since $\overrightarrow{R} - \overrightarrow{R_1}$ is a known value, we assume that n is a Gaussian white noise, that is, n is a normal distribution function with a mean of zero and a variance of σ^2. Therefore, the value in $\overrightarrow{\Delta R}$ obeys the normal distribution with mean R_{i1}^0 and variance σ^2. Since the measured values are independent, the likelihood function is

$$\prod_{i-1}^{M-1} \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{(\Delta R_i - R_i + R_1)^2}{2\sigma^2} \right\} \right]$$

$$\left[\frac{1}{\sqrt{2\pi}\sigma} \right]^{M-1} \exp\left(-\frac{\left(\overrightarrow{\Delta R} - \overrightarrow{R} - \overrightarrow{R_1}\right)^T \left(\overrightarrow{\Delta R} - \overrightarrow{R} - \overrightarrow{R_1}\right)}{2\sigma^2} \right) \tag{5}$$

It can be seen from the above equation that the coordinate value that requires the largest likelihood value is equivalent to

$$(x, y) = \arg\left\{\min\left[\left(\overrightarrow{\Delta R} - \overrightarrow{R} - \overrightarrow{R_1}\right)^T \left(\overrightarrow{\Delta R} - \overrightarrow{R} - \overrightarrow{R_1}\right)\right]\right\} \tag{6}$$

Since Equation (6) is a non-linear equation, if it is solved by the traditional analytical method, the solution will be accompanied by a complicated process and a huge amount of computation. In this paper, we use the cuckoo search algorithm to search for the optimal solution in the potential solution space to get the value of the point to be located.

3. Cuckoo Search Algorithm.

The cuckoo search algorithm is an algorithm that combines cuckoo brooding behavior with Levy flight. Because cuckoos find nests that are suitable for their own spawning in a random or similar way, in order to simulate cuckoo nesting, we first need to set three ideal states [5]: 1) State 1: Each cuckoo can only lay one egg at a time, and randomly choose to lay the nest; 2) State 2: Randomly select a group of nests, and keep the highest quality nests to the next generation; 3) Stage 3: The number of nests remains unchanged. If the host finds parasitic eggs of cuckoos (the probability of finding eggs is Pa $(0 < Pa < 1)$), a new nest will be established randomly and the old nest will be replaced. Cuckoo algorithm uses Levy flight to update bird's nest. The formula is as follows:

$$X_{t+1} = X_t + \alpha \otimes Levy(\beta) \tag{7}$$

where X_t denotes the location of the nest in the t generation, α denotes the step size scaling factor, $\otimes$ denotes the point multiplication $(\cdot\times)$ operation, and $Levy(\beta)$ denotes the Levy random path, and its probability density function is

$$Levy(\beta) \to \mu = t^{-\beta}, \quad (1 < \beta < 3) \tag{8}$$

$Levy(\beta)$ has infinite mean and infinite variance. This pattern can increase species diversity. In the early stage of the algorithm, we can search with a larger step size. In the later stage, we can use a smaller step size search to make it converge faster. The most efficient and straightforward way is to use the Mantegna algorithm to achieve a stable Levy distribution.

The formula for generating Levy random number in the algorithm [6] is as follows:

$$Levy(\beta) = \frac{\phi * \mu}{|v|^{1/\beta}} \tag{9}$$

where μ, v obey the standard normal distribution, β is usually taken as a constant of 1.5.

$$\phi = \left(\frac{\Gamma(1+\beta) * \sin\left(\pi * \frac{\beta}{2}\right)}{\Gamma\left(\left(\frac{1+\beta}{2}\right) * \beta * 2^{\frac{\beta-1}{2}}\right)}\right)^{\frac{1}{\beta}} \tag{10}$$

where

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt \tag{11}$$

According to the rules above, we can describe the cuckoo algorithm in steps.

1) Initialize various parameters of the algorithm, such as the number of nests, the probability of discovery Pa, and the number of iterations of the algorithm, etc.
2) Define the objective function, calculate the fitness function value of each bird's nest according to the fitness function, and record the current optimal nest position. According to Equation (6), the fitness function can be taken as:

$$Fit = \frac{1}{\left(\overrightarrow{\Delta R} - \overrightarrow{R} - \overrightarrow{R_1}\right)^T \left(\overrightarrow{\Delta R} - \overrightarrow{R} - \overrightarrow{R_1}\right)} \tag{12}$$

3) Retain the position of the contemporary optimal nest and update the rest of the nests according to Equation (7).
4) Calculate the value of the fitness function of the updated nest and compare it with the value of the fitness function before the update. If it is better than the nest before the update, replace it. If it is no better than the nest before the update, the pre-update is retained nest.
5) The nest host finds the cuckoo eggs with probability Pa, and randomly changes the position of the nest with Equation (7), thereby obtaining a new generation of nests.
6) Return to Step 2) to perform multiple iterations until the algorithm end condition is met.

4. **CS-BPNN Correction Model.**

BP neural network is a multi-layer feedforward neural network trained according to the error back propagation algorithm. It is the most widely used neural network. It has been proved that the single hidden layer neural network can simulate any continuous function [7]. In this paper, the amount of data we use is small, so the model is relatively uncomplicated. After experimental verification, we choose the structure of the single hidden layer neural network to build the model.

In this paper, we divide the positioning steps into an offline phase and an online phase. In the offline phase, we collect coordinates with and without NLOS error, respectively. The coordinates in the NLOS environment are obtained by using the cuckoo search algorithm. The points are divided into training sets and test sets according to 8 : 2. The BP neural network is trained with the training sets. In the online phase, the trained BP neural network is used to correct the test sets, thereby improving the positioning accuracy.

Figure 1 shows the CS-BP neural network model for coordinate correction. The BP neural network consists of three layers: the input layer, the hidden layer, and the output layer. For the input value coordinates of the network, we usually normalize it.

The input of the neural network is $p_j = [x, y]$ which is obtained from cuckoo search algorithm, the output of the neural network is $a_{2_k} = [a, b]$. The number of input layer neurons is 2; the number of neurons in hidden layer is n, the activation function between the input layer and the output layer is represented by f_1; the connection weight between the input layer and the output layer is represented as ω_{ij} and bias is represented as b_{1_i},

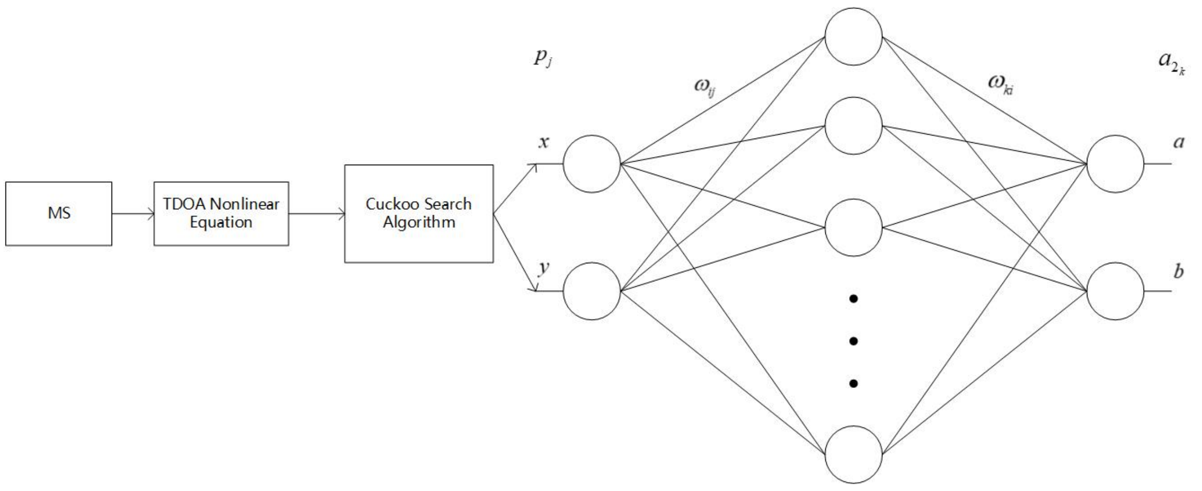

FIGURE 1. CS-BP neural network modified model

so the output of the i-th neuron of the hidden layer is

$$a_{1_i} = f_1\left(\sum_{j=1}^{2} \omega_{ij} p_j + b_{1_i}\right), \quad i = 1, 2, \ldots, n \tag{13}$$

The number of output layer neurons is 2, the activation function between the input layer and the output layer is represented by f_2; the connection weight between the input layer and the output layer is represented as ω_{ki} and bias is represented as b_{2_k}, so the output of the k-th neuron in the output layer is

$$a_{2_k} = f_2\left(\sum_{i=1}^{n} \omega_{ki} a_{1_i} + b_{2_k}\right), \quad k = 1, 2 \tag{14}$$

The loss function is

$$E = \frac{1}{2}\sum_{k=1}^{2}\left(t_k - a_{2_k}\right)^2 \tag{15}$$

Since E is a function of the weights ω_{ij} and ω_{ki}, the gradient descent method can be used to adjust the weights, thereby reducing the value of the loss function E. According to the gradient descent method, for the learning rate η, the change in weight is

$$\Delta\omega_{ki}(t) = -\eta\frac{\partial E}{\partial \omega_{ki}} = -\eta\frac{\partial E}{\partial a_{2_k}} \cdot \frac{\partial a_{2_k}}{\partial \omega_{ki}} = \eta(t_k - a_{2_k}) f_2' a_{1_i} \tag{16}$$

$$\Delta\omega_{ij}(t) = -\eta\frac{\partial E}{\partial \omega_{ij}} = -\eta\frac{\partial E}{\partial a_{2_k}} \cdot \frac{\partial a_{2_k}}{\partial a_{1_i}} \cdot \frac{\partial a_{1_i}}{\partial \omega_{ij}} = \eta\sum_{k=1}^{2}(t_k - a_{2_k}) f_2' \omega_{ki} f_1' p_j \tag{17}$$

The weights are adjusted according to the following formula:

$$\omega_{ki}(t+1) = \omega_{ki}(t) + \Delta\omega_{ki}(t) \tag{18}$$

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \Delta\omega_{ij}(t) \tag{19}$$

After the BP neural network model is trained according to the above process, the final result of the algorithm is

$$[a, b] = \mathrm{sim}(\mathrm{net}, [x, y]) \tag{20}$$

5. **Flow Chart of CS-BP Algorithm.** According to the steps of CS-BP algorithm, the flow chart of the algorithm is drawn, as shown in Figure 2.

6. **Simulation and Results Analysis.** In this paper the simulation software used in this article is MATLAB R2017b. We will simulate and verify the proposed algorithm in the UWB NLOS environment and compare it with the traditional algorithm. In the simulation, the position coordinates of our base station cell distribution are $(0, 0)$, $(-\sqrt{3}R, 0)$, $(\sqrt{3}R, 0)$, $(\sqrt{3}R/2, 3R/2)$, $(-\sqrt{3}R/2, -3R/2)$, $(-\sqrt{3}R/2, -\sqrt{3}R/2)$, $(\sqrt{3}R/2, -\sqrt{3}R/2)$. In this paper, R takes 15. The mobile station is randomly generated in the 1/12 cell around $(0, 0)$ base station, and we select 1000 points, 800 of which are training set and 200 points are test set.

In the cuckoo search algorithm, we set the number of nests to 50, the number of iterations to 150, and the probability of discovery (Pa) to 0.25. In the BP neural network, the number of input neurons is 2, the number of hidden neurons is 18, and the number of the output neurons is 2.

In order to verify the experimental results, we use the root mean square error and the cumulative distribution function to evaluate the performance of the algorithm model.

Figure 3 shows the simulation results of the root mean square error of Chan algorithm, Taylor algorithm, CS algorithm and CS-BPNN algorithm when the NLOS error is exponentially distributed with different mean values [8]. When the NLOS error is small, the positioning accuracy of the three positioning algorithms is equivalent. With the increase

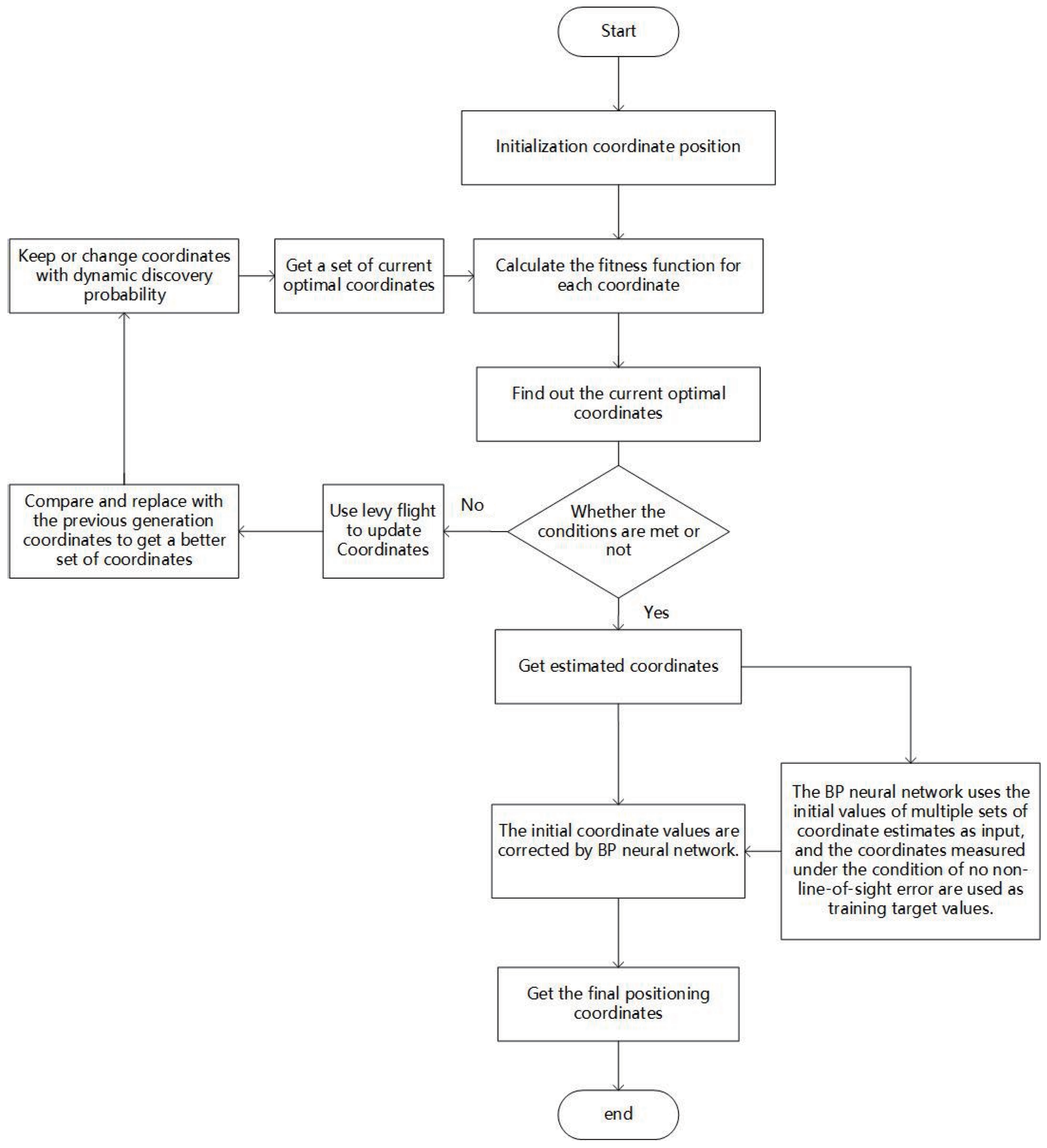
Start
Initialization coordinate position
Keep or change coordinates with dynamic discovery probability
Get a set of current optimal coordinates
Calculate the fitness function for each coordinate
Find out the current optimal coordinates
Compare and replace with the previous generation coordinates to get a better set of coordinates
Use levy flight to update Coordinates
No
Whether the conditions are met or not
Yes
Get estimated coordinates
The initial coordinate values are corrected by BP neural network.
The BP neural network uses the initial values of multiple sets of coordinate estimates as input, and the coordinates measured under the condition of no non-line-of-sight error are used as training target values.
Get the final positioning coordinates
end

FIGURE 2. CS-BP algorithm flow chart

of NLOS error, the proposed algorithm has a greater advantage than the Chan algorithm, Taylor algorithm and the CS algorithm, which reflects the good inhibition of NLOS error.

Figure 4 shows the error cumulative distribution function of each algorithm when the NLOS mean is 0.8 m. For the sake of observation, we only take 10 points for comparison. It can be found that the proposed algorithm reaches probability 1 faster than the other three algorithms, and the error is a stable distribution between 0.03 m and 0.18 m. The result of the cuckoo search algorithm is the most stable, so it can provide good data for BPNN. From this point of view, the CS-BPNN algorithm has excellent positioning performance.

In order to determine the range of positioning accuracy of the improved algorithm, we conducted several independent and repeated experiments. The simulation result of the positioning accuracy is shown in Figure 5 under the experiments.

The simulation results of MATLAB show that the accuracy of the algorithm is mostly stable between 0.07 m and 0.13 m, whether it is repeated 5 times or 10 times or even more.

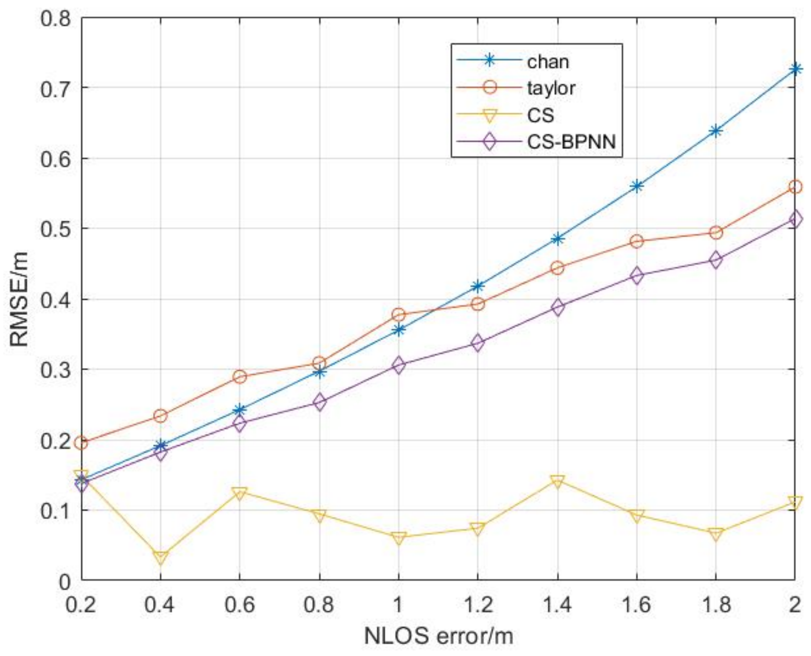

FIGURE 3. RMSE comparison between the improved algorithm and the traditional algorithm

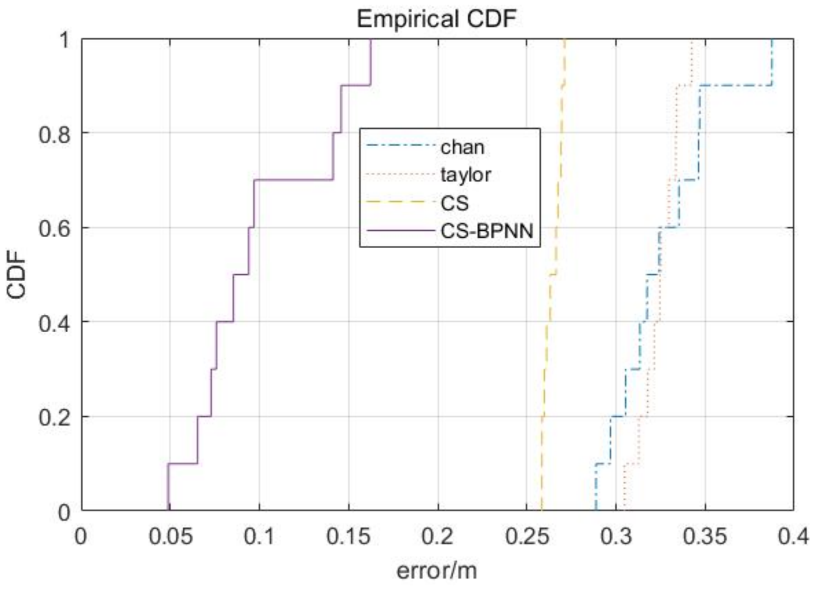

FIGURE 4. CDF – Estimation error

The experimental results show that the proposed location algorithm has high stability and can meet our needs for tag location.

7. **Conclusions.** The cuckoo search algorithm is a new type of intelligent search algorithm. This paper mainly studies an improved TDOA algorithm based on CS-BPNN and applies it to the UWB system. In this paper, the cuckoo search algorithm is used to estimate the nonlinear equation of TDOA, and then the estimated value is used as input to train BP neural network. Finally, the trained neural network is used to correct the estimated value of the cuckoo search algorithm. The simulation results show that the proposed algorithm has significantly improved relocation accuracy compared with the traditional algorithm and CS algorithm, and its positioning accuracy is approximately

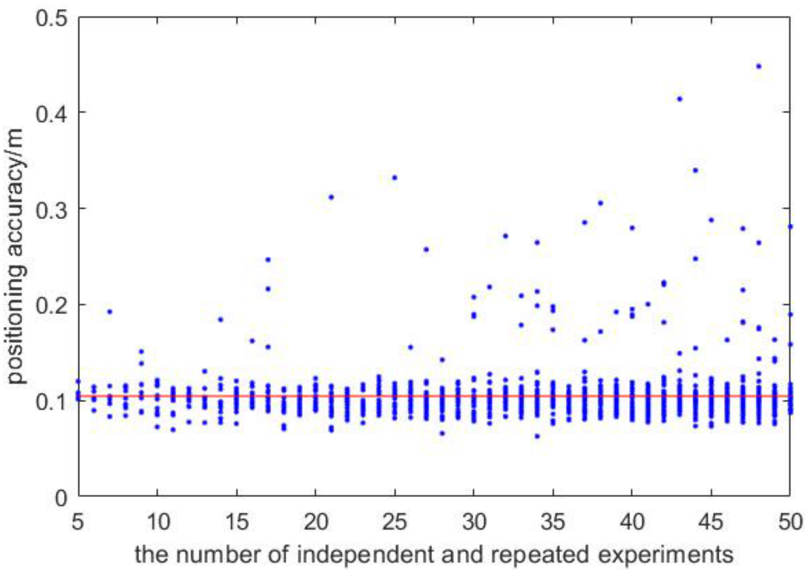

FIGURE 5. Multiple independent replicate experiments of improved algorithm

maintained between 0.07 m ∼ 0.13 m. However, when the environment changes, the algorithm needs to recalculate the data and train the BP neural network. In the future, we will further improve the robustness of the algorithm.

Acknowledgment. This research was financially supported by the National Natural Science Foundation of China (Nos. 61461017, 61861015); the Hainan Natural Science Foundation Innovation Research Team Project (No. 2017CXTD0004); the Key Project of Scientific Research of Higher Education in Hainan Province (No. Hnky2019ZD-35).

REFERENCES

[1] K. Zhang, C. Shen, H. Wang, Q. Gao, H. Li and N. Li, An improved three-dimensional location algorithm and simulation of AOA and TDOA based on wave interference sensors, *Boletin Tecnico/Technical Bulletin*, vol.55, no.19, pp.211-219, 2017.

[2] L. Gao, H. Sun, M. Liu and Y. Jiang, TDOA collaborative localization algorithm based on PSO and Newton iteration in WGS-84 coordinate system, *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, Chengdu, pp.1571-1575, 2016.

[3] J. Li, L. Zhang and H. Ma, A new positioning method combining GA and iterative LS algorithm in NLOS environment, *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, Guangzhou, pp.403-407, 2017.

[4] H. Shi and J. Cao, A new hybrid algorithm on TDOA localization in wireless sensor network, *2011 IEEE International Conference on Information and Automation*, Shenzhen, pp.606-610, 2011.

[5] X. S. Yang and S. Deb, Multiobjective cuckoo search for design optimization, *Computers & Operations Research*, vol.40, no.6, pp.1616-1624, 2013.

[6] X. S. Yang and S. Deb, Engineering optimization by cuckoo search, *Int'l Journal of Mathematical Modeling and Numerical Optimization*, vol.1, no.4, pp.330-343, 2010.

[7] M. T. Hagan, *Neural Network Design*, PWS, 1999.

[8] B. Li, W. Cui and B. Wang, A robust wireless sensor network localization algorithm in mixed LOS/NLOS scenario, *Sensors*, vol.15, no.9, pp.23536-23553, 2005.

[9] K. Zhang, C. Shen, Q. Gao, L. Zheng, H. Wang and Z. Li, Precise positioning system of ship interior based on UBW ultra wideband technology, *Journal of Coastal Research*, vol.2018, no.S83, pp.908-912, 2018.

[10] F. Dong, C. Shen and K. Zhang, Real-valued DOA estimation for MIMO array system under unknown nonuniform noise, *IEEE Access*, vol.6, no.1, pp.52218-52226, 2018.