Cooperative Localization Algorithm based on Coverage Optimization of Actors for Wireless Sensor and Actor Networks

Huanliang Xu, Duo Li, Jianhua Che, and Haoyun Wang*
College of Information Science and Technology, Nanjing Agricultural University, Nanjing, China
*Corresponding author, Email: huanliangxu@njau.edu.cn, 2011114004@njau.edu.cn, chejianhua@njau.edu.cn, wanghy@njau.edu.cn

***Abstract*—Different from existing range-free algorithms in wireless sensor networks (WSN), a cooperative localization algorithm based on coverage optimization of actors (CLCOA) for wireless sensor and actor networks (WSAN) is proposed. It uses mobile actors instead of anchors in WSN. Firstly, the area of the unknown node is determined through the movement of positioning actors. Then this area decreases through iteration. When localization accuracy is satisfied, the centroid of this area is calculated and treated as the coordinate of the unknown node. Free actors adjust their positions through virtual force while locating actors work. Accordingly, actors' coverage is optimized. Via simulation, it is proven that CLCOA has high locating accuracy with RSSI error and GPS error, and the introduction of virtual force improves actors' coverage and locating speed.**

***Index Terms*—WSN; WSAN; Mobile Actors; Range-Free Localization; Virtual Force**

I. INTRODUCTION

The development of WSAN is on the basis of WSN. WSAN consists of a group of sensor nodes and actor nodes, sensor nodes are used to acquire information from environment, and actors are responsible to change environment [1]. Different from WSN, WSAN can change the physical world. Since WSAN is proposed, many research points in this field have received great attention among researchers, including localization problem. Acquiring position imformation of sensor nodes is very important for many WSAN applications, because the sensed data without position information of sensor nodes is meaningless [2]. For instance, sensor nodes can get the motion area of hostile vehicle to make precise battle strategy in battlefield. Moreover, it assists in the management and operation of network. Currently, classification methods of WSN localization algorithms mainly include: range-based localization and range-free localization, distributed localization and centralized localization, absolute localization and relative localization, anchor-based localization and anchor-free localization. Because of the introduced actors, existing WSN localization algorithms cannot be used directly in WSAN.

In this paper, we propose a cooperative localization algorithm based on coverage optimization of actors for wireless sensor and actor network (CLCOA). Positioning actors form a square area to localize sensor node while free actors adjust their positions by virtual force, so actors are well distributed and coverage is optimized. It guarantees that network events can be disposed in time, thus improving task execution efficiency of the network. CLCOA uses a few mobile actors instead of anchors in WSN to save the deployment cost, and this makes free actors well distributed by combining virtual force model.

The remainder of the paper is organized as follows: Section 2 describes the related work. Section 3 presents CLCOA localization algorithm in detail. Simulation and analysis are discussed in Section 4. Finally, Section 5 gives the conclusions.

II. RELATED WORK

Anchors are generally used to assist in localizing sensor nodes in WSN, which are aware of their location through GPS or manual deployment. Most commonly, according to whether distance is measured, WSN localization algorithms can be classified into two categories: range-based and range-free. Received signal strength indication (RSSI) [3], time of arrival (TOA) [3] and time difference of arrival (TDOA) [3] are typical measurement techniques. The majority of them have high demand for hardware, like installing special antenna for nodes, thus not fitting for large-scale network. Limited transmission distance of signal and non line-of-sight conditions might affect the distance estimates. In addition, these techniques use varieties of methods to reduce localization error, which leads to a large number of computing and communication overheads [4]. Range-free solutions [3-5] include centroid algorithm, approximate point in triangulation test (APIT), convex position estimation, and distance vector hop (DV-Hop). They require no additional hardware and have low costs. But the accuracy is relatively low, and their performance depends on the network topology [6].

Since WSN localization algorithms can't be simply transplanted to WSAN, localization in WSAN has become a key problem to solve. Some progress has been made in research of localization in WSAN so far. An

doi:10.4304/jnw.9.5.1145-1156

efficient cooperative localization scheme (ECLS), which is an event-driven method, is proposed by Han P et.al [7]. It only localizes the position of the node detecting the wanted event and uses actors with unconstrained resource to assist in localizing sensor nodes. Therefore, energy consumption of the system can be effectively reduced. It is theoretically simple and uses RSSI to measure distance. But there are also some inevitable shortages for RSSI, like unstable signal, low accuracy and so on. The algorithm proposed by Giacomo G [8] is the first semi-distributed protocol featuring asynchronous sleep/awake schedules. It organizes the sensors near each actor by means of a discrete polar coordinate system. Each sensor is localized when both the corona and sector coordinates are acquired. Based on range-free method, the algorithm belongs to the class of coarse-grained localization protocols, so the accuracy cannot be guaranteed. Mustafa I A et.al [9] presents a solution for mobile sensor nodes with the goal of monitoring the Amazon River. Using a multi-hop approach, it is scalable and is fit for the large-scale network. The drawback is the initial weight value of sensor has a great effect on accuracy, and localization accuray will decrease as the value goes up. In order to localize the mobile sensor nodes relative to the actors, a novel Timing-based Mobile Sensor Localization (TMSL) algorithm is introduced [10]. It uses TOA to estimate distance. The most distinguished is that it requires no time synchronization among the sensor nodes or with the actors. In TMSL, propagation time of the beacons is computed via a special time model. However, excessively detailed time division leads to high computation complexity. An event driven localization scheme based on the RSSI range measuring for dynamic WSAN is proposed [11], and it is just similar to RSSI. But the accuracy of RSSI can't meet high demand in WSAN. So it can be concluded that original RSSI is not suitable for WSAN localization. Localization techniques based on semidefinite programming are researched in [12]. Novel algorithms are respectively proposed in range-based and range-free mode. Here, locating problems are translated into convex optimization, and many mathematic formulas are used to compute. Moreover, via simulation experiments, they are proven to have high locating accuracy and computing complexity. ADAPTPLACDVDIST [13] is proposed for localizing nodes in NLOS environment, which is a class of DV-Hop. It moves the localizers in each of the multi-hop chains, thus minimizing the multi-hop distance between the references and the node. The drawback is that the solution requires large number of localizers and its localization accuracy depends on the number of localizers in each chains.

Existing WSAN localization algorithms may cause uneven distribution of actors during localization. The density of actors may be too close or sparse in local area, so actors may not react to the reported event in other areas in time. These problems can be well solved by virtual force. Initially, virtual force is used to make robots avoid barriers, and then used in distribution optimization of network [14]. At present, virtual force model is introduced to the research on nodes localization and coverage in WSN. In [15], a distributed actor deployment algorithm for maximum connected coverage (DA^{2}MC) in WSAN is proposed. This paper proves that regular six polygon can be used to achieve maximum coverage with least waste. DA^{2}MC moves actor nodes by virtual repelling force to extend their coverage while maintaining sensor-actor connectivity. In [14], a multi-hops localization algorithm based on virtual force for WSN is proposed. The number of beacon messages accepted by unknown node is used as a parameter of the force calculation. By optimizing the network distribution, it is more reasonable to calculate and select the correction value. A coverage algorithm based on virtual grids area density is proposed in [16], and particular swarm optimization (PSO) localization algorithm is further improved to realize effective coverage of sensor nodes while improving the localization accuracy.

To solve the problems above, CLCOA is proposed to realize localization in WSAN. It uses a few mobile actors instead of anchors, and overcomes the shortcomings of range-free location algrithoms in WSN, which seriously depend on density of anchors. For example, it avoids erroneous judgments in APIT, and is better than DV-Hop in the respect of computing and communication. As to [7] and [11], they just make improvements or combine other algorithms based on RSSI, however, the thought of area iterative refinement in CLCOA are obviously better in localization accuracy. The algorithm in [9] is similar to DV-Hop, and static actors make its applications limited. Instead, CLCOA uses mobile actors. In addition, CLCOA refers to the thought of virtual force in [14-16], during locating sensor nodes, it makes actors well distributed and optimizes coverage. Accordingly, localization time and overhead are reduced.

III. Definition of Virtual Force

We assume that sensor nodes are static and resource-limited. They are responsible to monitor the network area and report the event to actor nodes. Actor nodes, which are mobile and powerful, take related action according to the reported data. Actor nodes are also equipped with GPS and aware of their positions at any time. In addition, actor nodes are classified into two categories: positioning actors and free actors.

A. Virtual Force Model

Actor nodes can be classified into two categories according to whether or not they take part in localizing the sensor node:

a_p: positioning actors

a_f: free actors

Since a_p take part in localizing the sensor node, their virtual force are ignored. We assume that virtual force only acts on a_f from a_p and a_f. The force shows up as vectorial repelling force or attractive force, and the magnitude of force is related to the types of nodes and the distance between them.

So if node j exerts a force on node i, $\vec{F}_{ij}$ can be confirmed by formula (1):

$$\vec{F}_{ij}=\begin{cases} k_{r\_p}(\frac{1}{d_{ij}}-\frac{1}{r_{th}}) & if \quad 0<d_{ij}\le r_{th}, i\in a_f, j\in a_p \\ k_{r\_f}(\frac{1}{d_{ij}}-\frac{1}{r_{th}}) & if \quad 0<d_{ij}\le r_{th}, i\in a_f, j\in a_f \\ k_{a\_p}(d_{ij}-r_{th}) & if \quad r_{th}<d_{ij}\le r_c, i\in a_f, j\in a_p \\ k_{a\_f}(d_{ij}-r_{th}) & if \quad r_{th}<d_{ij}\le r_c, i\in a_f, j\in a_f \\ 0 & else \end{cases} \quad (1)$$

In the above formula, k_{r_p}, k_{r_f}, k_{a_p} and k_{a_f} are virtual force coefficients which conform to $k_{r\_p}>k_{r\_f}$ and $k_{a\_p}>k_{a\_f}$, d_{ij} is the distance between the two nodes, r_{th} is distance threshold, and r_c is the one hop communication radius. If $d_{ij}\le r_{th}$, $\vec{F}_{ij}$ shows up as the repelling force. And if $r_{th}<d_{ij}\le r_c$, $\vec{F}_{ij}$ shows up as the attractive force. Otherwise, $\vec{F}_{ij}=0$.

The calculated force $\vec{F}_i$ is the vectorial addition of these virtual forces, and can be expressed as formula (2):

$$\vec{F}_i=\sum_{i\ne j}\vec{F}_{ij} \quad (2)$$

B. Adjustment of Node Position

Each node moves to a new position according to $\vec{F}_i$, the new coordinates can be computed as follows:

$$x'=\begin{cases} x & if \quad |F_i|\le F_{th} \\ x+\frac{F_x}{|F_i|}\times Smax & if \quad |F_i|>F_{th} \end{cases} \quad (3)$$

$$y'=\begin{cases} y & if \quad |F_i|\le F_{th} \\ y+\frac{F_y}{|F_i|}\times Smax & if \quad |F_i|>F_{th} \end{cases} \quad (4)$$

In the above formulas, F_x and F_y are components of $\vec{F}_i$ in X-axis and Y-axis, Smax is the maximum range that the actor can move at every time, F_{th} is virtual force threshold, and the actor will not move if $\vec{F}_i<F_{th}$.

IV. Proposed Scheme

In this part, a cooperative localization algorithm based on coverage optimization of actors for WSAN is proposed. Fig. 1 presents the main process of CLCOA. Detailed steps of this scheme will be discussed as follows.

A. Design of CLCOA

1) Classification of Actor Nodes

The sensor node S starts to send a request, and then selects 5 actor nodes which firstly reply to take part in localizing. These actor nodes are set to a_p, a_p={A_0, A_1, A_2, A_3, A_4}, and other actors are set to a_f. Each actor node keeps a neighbor table to record the coordinates of its neighbor nodes.

2) Formation of Square Area

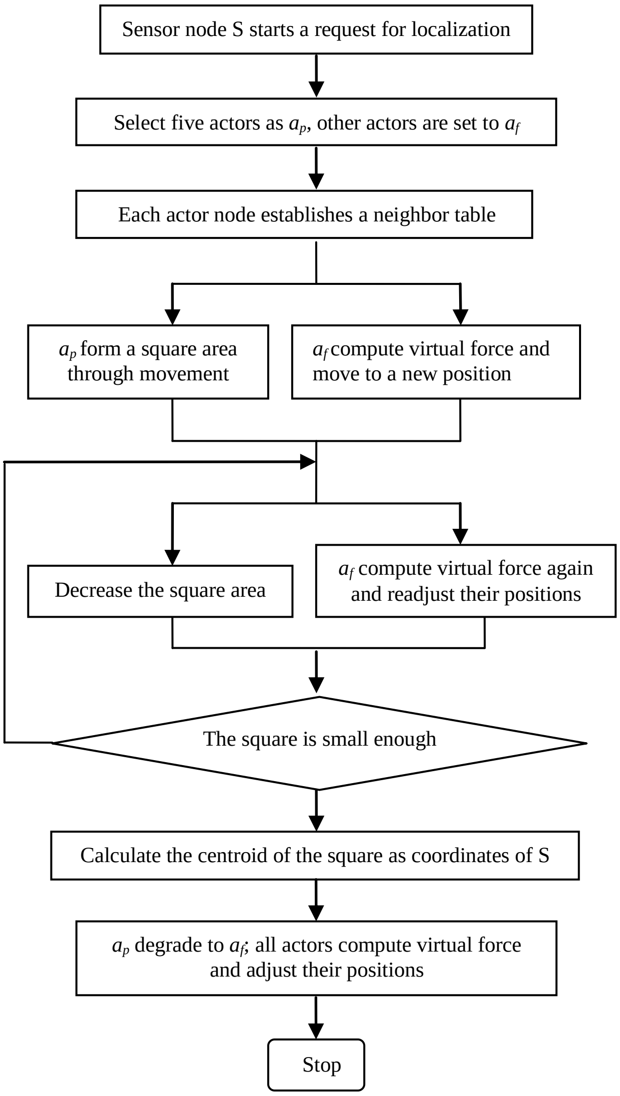

Figure 1. Flowchart of CLCOA

a_p form a square area through movement, which makes sensor S in this square area. The process is: actor A_0 firstly moves close to sensor S, and in the course of moving, it discontinuously sends RF signal to sensor S. Sensor S computes received signal strength [17-18], and it sends a feedback signal to actor A_0 when signal strength meets the threshold a, which is to say, the distance between two nodes is very small, and $RSSI(d_{as})\ge a>RSSI(R_s)$. When actor A_0 receives the signal, it stops moving.

Actor A_0 stops and broadcasts its position information (X_0, Y_0). Actor A_1, A_2, A_3 and A_4 adjust their positions according to (X_0, Y_0) and broadcast their current coordinates, thus forming a square area as shown in Fig.2. Moreover, expecting coordinates of these four actors can be computed as formulas (5-8).

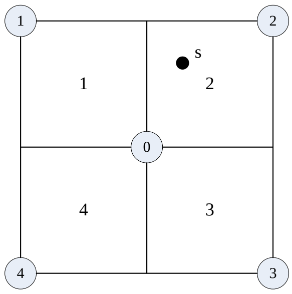

Figure 2. Positional relationship of a_p

$$(X_1,Y_1)_{A_1} = (X_0 - d_{as}, Y_0 + d_{as}) \tag{5}$$

$$(X_2,Y_2)_{A_2} = (X_0 + d_{as}, Y_0 + d_{as}) \tag{6}$$

$$(X_3,Y_3)_{A_3} = (X_0 + d_{as}, Y_0 - d_{as}) \tag{7}$$

$$(X_4,Y_4)_{A_4} = (X_0 - d_{as}, Y_0 - d_{as}) \tag{8}$$

3) Redistribution of Free Actors

Through the above steps, a_p forms a square which the length of a side is $2d_{as}$, ensuring that sensor S must be in it. At the same time, a_f compute virtual force according to steps (1-2) and move to a new position. So actors in the network are redeployed.

4) Decreasing Square Area Through Iteration

Sensor S further judges the subregion. The method is that sensor S receives RF signals from actor A_1, A_2, A_3 and A_4, and records these signals to compare. The rule is as follows:

If $RSSI_1$ is the maximum value and $RSSI_3$ is the minimum value, then sensor S is in region 1

If $RSSI_2$ is the maximum value and $RSSI_4$ is the minimum value, then sensor S is in region 2

If $RSSI_3$ is the maximum value and $RSSI_1$ is the minimum value, then sensor S is in region 3

If $RSSI_4$ is the maximum value and $RSSI_2$ is the minimum value, then sensor S is in region 4

If sensor S is on the boundary, signal strength from the two actors in two sides is equal in theory. At this moment, the judgement of subregion has no effect on localization, so it can be processed randomly.

After the subregion is confirmed, sensor S sends RF signal to A_0. For example, sensor S confirms that it is in region 2, and it sends a signal to A_0. After A_0 receives the signal, it computes the center coordinate of region 2. Then, it moves to this position and broadcasts its current coordinates. After receiving coordinates information from A_0, actor A_1, A_2, A_3 and A_4 compute new coordinates according to formulas (9-13), adjust their positions and broadcast current coordinates. At this moment, a_p move to new position, and a_f compute virtual force again and readjust their positions.

$$(x,y)_{A_0} = \begin{cases} (X_0 - \frac{1}{4}d_i, Y_0 + \frac{1}{4}d_i) & S \in \Omega_1 \\ (X_0 + \frac{1}{4}d_i, Y_0 + \frac{1}{4}d_i) & S \in \Omega_2 \\ (X_0 + \frac{1}{4}d_i, Y_0 - \frac{1}{4}d_i) & S \in \Omega_3 \\ (X_0 - \frac{1}{4}d_i, Y_0 - \frac{1}{4}d_i) & S \in \Omega_4 \end{cases} \tag{9}$$

$$(x,y)_{A_1} = \begin{cases} (X_1, Y_1) & S \in \Omega_1 \\ (x_{A_0} - \frac{1}{4}d_i, y_{A_0} + \frac{1}{4}d_i) & S \in \Omega_2 \\ (x_{A_0} - \frac{1}{4}d_i, y_{A_0} + \frac{1}{4}d_i) & S \in \Omega_3 \\ (x_{A_0} - \frac{1}{4}d_i, y_{A_0} + \frac{1}{4}d_i) & S \in \Omega_4 \end{cases} \tag{10}$$

$$(x,y)_{A_2} = \begin{cases} (x_{A_0} + \frac{1}{4}d_i, y_{A_0} + \frac{1}{4}d_i) & S \in \Omega_1 \\ (X_2, Y_2) & S \in \Omega_2 \\ (x_{A_0} + \frac{1}{4}d_i, y_{A_0} + \frac{1}{4}d_i) & S \in \Omega_3 \\ (x_{A_0} + \frac{1}{4}d_i, y_{A_0} + \frac{1}{4}d_i) & S \in \Omega_4 \end{cases} \tag{11}$$

$$(x,y)_{A_3} = \begin{cases} (x_{A_0} + \frac{1}{4}d_i, y_{A_0} - \frac{1}{4}d_i) & S \in \Omega_1 \\ (x_{A_0} + \frac{1}{4}d_i, y_{A_0} - \frac{1}{4}d_i) & S \in \Omega_2 \\ (X_3, Y_3) & S \in \Omega_3 \\ (x_{A_0} + \frac{1}{4}d_i, y_{A_0} - \frac{1}{4}d_i) & S \in \Omega_4 \end{cases} \tag{12}$$

$$(x,y)_{A_4} = \begin{cases} (x_{A_0} - \frac{1}{4}d_i, y_{A_0} - \frac{1}{4}d_i) & S \in \Omega_1 \\ (x_{A_0} - \frac{1}{4}d_i, y_{A_0} - \frac{1}{4}d_i) & S \in \Omega_2 \\ (x_{A_0} - \frac{1}{4}d_i, y_{A_0} - \frac{1}{4}d_i) & S \in \Omega_3 \\ (X_4, Y_4) & S \in \Omega_4 \end{cases} \tag{13}$$

In the above formulas, $(x_{Ai}, y_{Ai})_{,i=0,1,2,3,4}$ is the expecting coordinates of actor A_i, $(X_i, Y_i)_{,i=0,1,2,3,4}$ is the current coordinate of actor A_i, d_i is the length of a side of the current square, and i is the current number of iteration.

By repeating the above steps to judge the subregion of sensor S, the square area can be reduced to a small range through iteration. When the square is small enough, the iteration stops and the centroid of the square is calculated and treated as coordinates of sensor S.

5) Coverage of Actors Based on Virtual Force

After localization is finished, a_p degrade to a_f. All actors compute virtual force again according to formulas (1-2) and adjust their positions. It makes actors well distributed in the network.

B. Implementation Mechanism

In reality, errors caused by actors' moving will affect the accuracy of CLCOA. To solve this problem, slope correction mechanism and position adjustment mechanism are proposed to optimize actors' moving.

(1) Slope correction mechanism

During localization, each actor always needs to move from its current coordinate to the expecting coordinate. Assume that the expecting coordinate is (X_n, Y_n), the current coordinate is (X_0, Y_0), we can obtain that the slope $k_n=(Y_n-Y_0)/(X_n-X_0)$, and the distance $D=\sqrt{(X_n-X_0)^2+(Y_n-Y_0)^2}$. The actor just need to move D along k_n. However, errors of movement may lead to difference between the real coordinate and the expecting coordinate, which is difficult to avoid. Therefore, CLCOA periodically corrects the slope in the course of movement. That is to say, the slope is recalculated once the actor moves some distance. So if error exists, the slope is corrected. Next, slope correction is discussed in detail.

The number of correction is computed as $Num = k \times (\sqrt{(Y_n - Y_0)^2 + (X_n - X_0)^2} - b)$, where both k and b are constants, in which k is a coefficient related to the distance and b is the minimum distance to adjust the slope. Moreover, both of them should be set according to the real condition. *Num* should be rounded down if it is a decimal.

Let (X_i, Y_i) be the coordinate of the actor after the *i*th move. D_i is the distance that the actor moves at the *i*th time, and it can be computed as $D_i = \sqrt{(X_n - X_{i-1})^2 + (Y_n - Y_{i-1})^2} / (Num - (i-1))$, $i = 1, 2 ... Num$. Let k_i be the slope after the *i*th move, and k_i can be computed as $k_i = (Y_i - Y_0)/(X_i - X_0)$. The actor corrects k_i after it moves D_i. $\Delta\theta$ is an angle that represents the difference between k_i and k_n, and it can be expressed as $\Delta\theta = |\arctan k_i - \arctan k_n|$. If $\Delta\theta > \sigma$, in which σ is the threshold of angle, k_i must be corrected or else the actor will go on moving along k_i.

(2) Position adjustment mechanism

After the actor moves to (X_n, Y_n), position adjustment is started to adjust coordinate of the actor. Here, we limit the error range to a radius of error *R*, as shown in Fig.3. If only the real coordinate of the actor is within the radius of (X_n, Y_n), the error can be ignored, and then the actor stops moving.

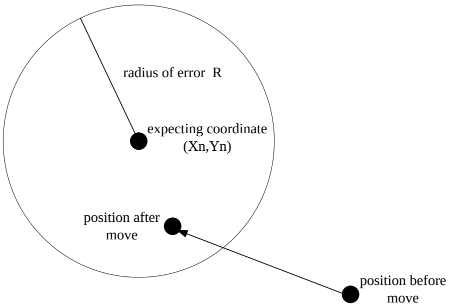

Figure 3. Illustration of radius of error

V. SIMULATION AND ANALYSIS

In this section, simulation experiments implemented in Java are used to evaluate the performance of CLCOA. The environment is as follows: Intel Core 2 Duo processor, 1GB of RAM, Windows XP, and Eclipse as compiler. The scenario is in a field of 500m×500m and 200 sensors and several actors are randomly deployed. Sensors are static and communication range is 20m. Actors are mobile and the speed is 10m/s. Communication range of actors are initially set to 50m.

Four simulation experiments are conducted to verify the algorithm validity. Experiment 1 evaluates the impact of RSSI threshold *a*, number of actors N_a, communication range of actors R_a, virtual force threshold F_{th} and virtual force coefficient k_{r_p} on CLCOA. As the four virtual force coefficients are related and only k_{r_p} is considered here, experiment 2 verifies the impact of virtual force model introduced on performance. Experiment 3 compares the performance of CLCOA with centroid algorithm [5]. Experiment 4 compares the performance of CLCOA with the localizing method of single actor. The method of single actor means the data acquired from GPS receiver on the actor, which is close to the sensor, is treated as the coordinate of the sensor. The results are the average of multiple experiments.

A. Performance Evaluation Index

We design four evaluation indexes to evaluate the performance of CLCOA.

1) Average localization error

Average localization error [19] can be expressed as follow:

$$\varepsilon = \frac{\sum_{i=1}^{n} \sqrt{(x_i - xr_i) + (y_i - yr_i)}}{n} \quad (14)$$

where (xr_i, yr_i) is the real coordinate of the *i*th sensor, (x_i, y_i) is the coordinate calculated by CLCOA of the *i*th sensor, and *n* is the number of sensors.

2) Average Localization Time

Average localization time can be expressed as:

$$t = \frac{t\_network}{n} \quad (15)$$

where *t_network* is the total runtime of the network, *n* is the number of sensors that finish localization within *t_network*. Localization time includes time of movement, transmission time and processing time.

3) Average Localization Overhead

Average localization overhead can be expressed as:

$$o = \frac{o\_total}{n} \quad (16)$$

where *o_total* is the total overhead that sensors cost to finish localization within *t_network*, and *n* is the number of sensors. Here, we count the number of packets to represent the overhead.

4) Coverage of Actors

Coverage of actors can be expressed as:

$$c = \frac{\bigcup_{i=1}^{n} A_i}{A} \text{ [14]} \quad (17)$$

where A_i is the coverage of the *i*th actor and *A* is the network area.

B. Simulation Experiment 1

The impact of *a*, N_a and other parameters on the performance of CLCOA are analyzed in this part. If not mentioned, the parameters take default values in the following experiments as below: *a*=-70dBm, N_a=14, R_a=50m, F_{th}=0.01, k_{r_p}=0.5, k_{r_f}=0.8×k_{r_p}, k_{a_p}=0.02, k_{a_f}=0.016.

1) The Impact of a

Fig. 4-Fig. 7 show the impact of *a* on *ε*, *t*, *o* and *c*.

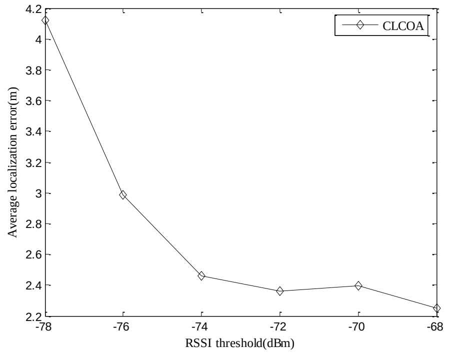

Figure 4. Impact of *a* on average localization error

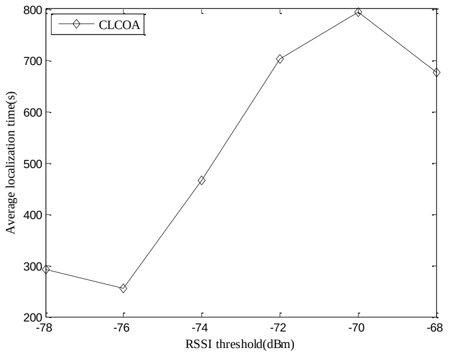

Figure 5. Impact of *a* on average localization time

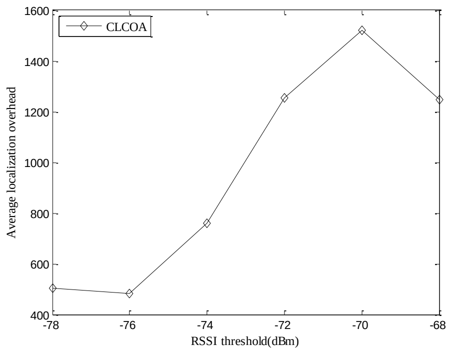

Figure 6. Impact of *a* on average localization overhead

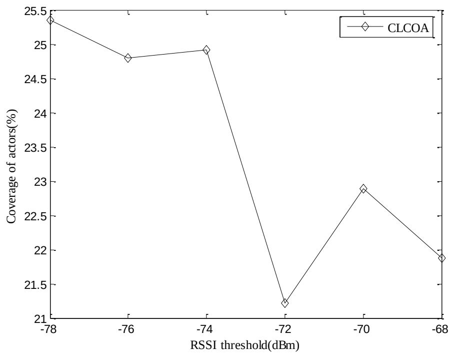

Figure 7. Impact of *a* on coverage of actors

It can be observed that ε decreases as the value of *a* increases, and the slope of curve gradually decreases in overall, which means the speed of decreasing slows down. When *a* equals to 68dBm, *ε* is about 2.252m. There is a growing tendency for *t* and *o* in overall, and this means decrease of *ε* causes growth of *t* and *o*. When *a* equals to -68dBm, *t* is 676s and *o* is 1248 packets. *c* changes little and fluctuates between 21% and 26%.

With the increasing of *a*, actor A_0 needs to moves more close to sensor S, which leads to growing time and overhead, and the initial square is smaller, so ε decreases.

2) The impact of N_a

Fig. 8-Fig. 11 show the impact of N_a on *ε*, *t*, *o* and *c*.

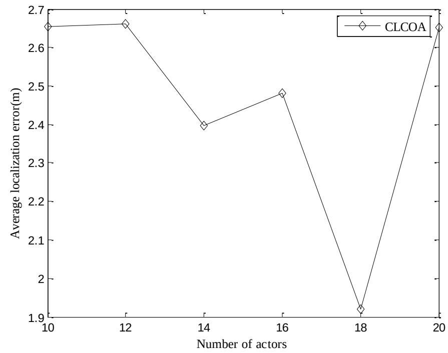

Figure 8. Impact of N_a on average localization error

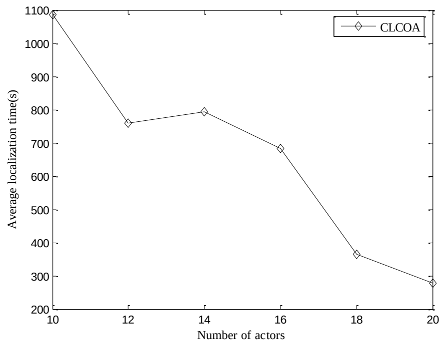

Figure 9. Impact of N_a on average localization time

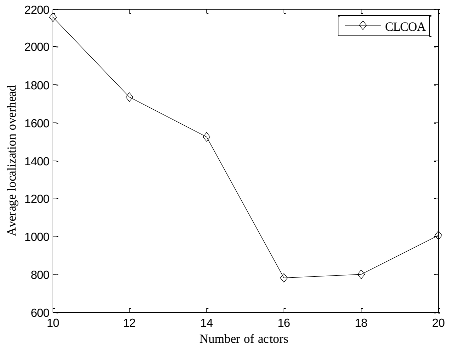

Figure 10. Impact of N_a on average localization overhead

It can be observed that *ε* changes little as the value of N_a increases and fluctuates between 1.9m and 2.7m. When N_a equals to 18, *ε* is about 1.920m. There is a declining tendency for *t* and *o* in overall. When N_a equals to 20, *t* is 278s and *o* is 1003 packets. *c* increases and the slope starts to decrease when N_a is more than 14, which means the increasing speed of *c* slows down.

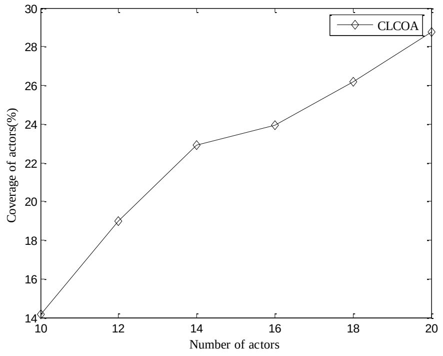

Figure 11. Impact of N_a on coverage of actors

With the increasing of N_a, the number of selectable actors to localize increases, which realizes concurrently localizing for multiple sensors. So *t* and *o* decreases.

3) The Impact of R_a

Fig. 12-Fig. 15 show the impact of R_a on *ε*, *t*, *o* and *c*.

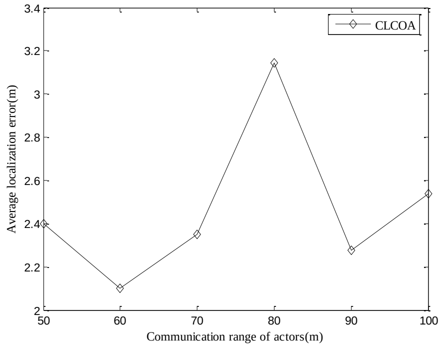

Figure 12. Impact of R_a on average localization error

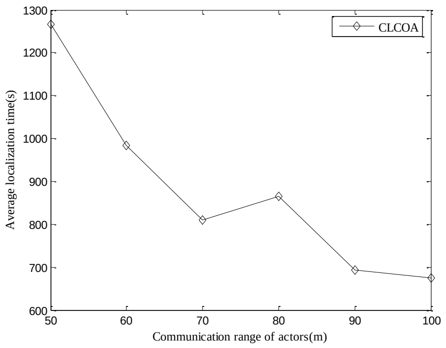

Figure 13. Impact of R_a on average localization time

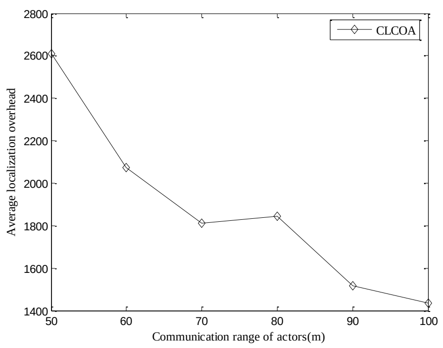

Figure 14. Impact of R_a on average localization overhead

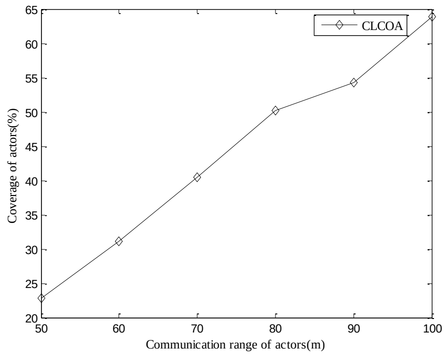

Figure 15. Impact of R_a on coverage of actors

It can be observed that *ε* varies within 1m as the value of R_a increases. There is a declining tendency for *t* and *o* in overall. When R_a equals to 100m, *t* is 674s and *o* is 1436 packets. *c* obviously increases and when R_a equals to 100m, *c* is 63.85%.

As R_a increases, packets are less forwarded, so *t* and *o* are reduced.

4) The Impact of F_{th}

Fig. 16-Fig. 19 show the impact of F_{th} on *ε*, *t*, *o* and *c*.

It can be observed that *ε* and *c* change little as the value of F_{th} increases. There is a growing tendency for *t* and *o*. When F_{th} equals to 0.01, *t* is 794s and *o* is 1522 packets.

As F_{th} increases, actors are less apt to move, which more easily leads to close distribution of actors in local. At this moment, if a sensor in sparse area starts a request, it will cost more time and overhead for actors to move close to the sensor and localizing it.

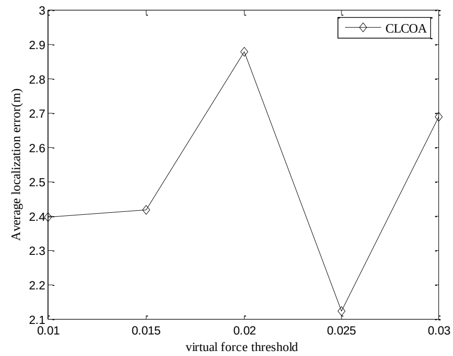

Figure 16. Impact of F_{th} on average localization error

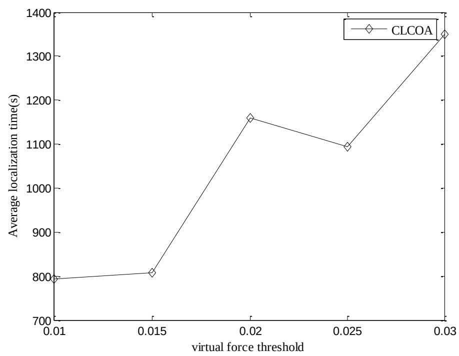

Figure 17. Impact of F_{th} on average localization time

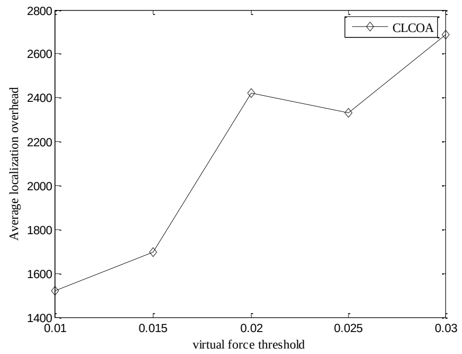

Figure 18. Impact of F_{th} on average localization overhead

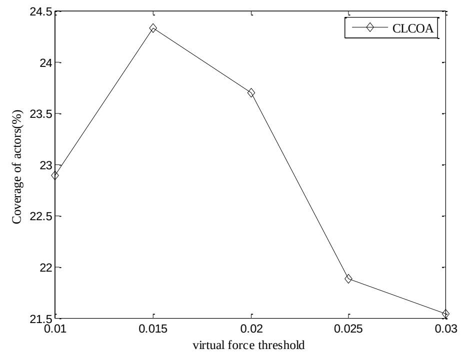

Figure 19. Impact of F_{th} on coverage of actors

5) The Impact of k_{r_p}

Fig. 20-Fig. 23 show the impact of k_{r_p} on ε, t, o and c.

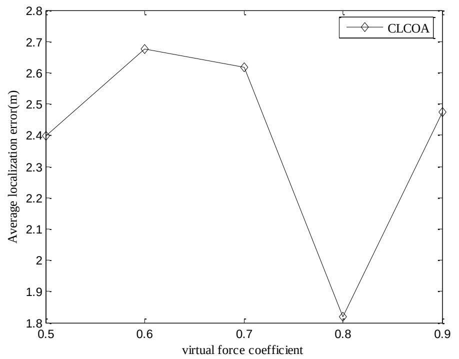

Figure 20. Impact of k_{r_p} on average localization error

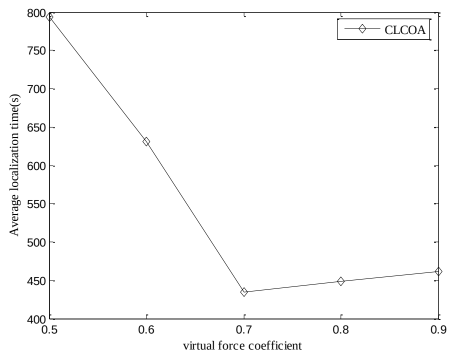

Figure 21. Impact of k_{r_p} on average localization time

It can be observed that ε and c change little as the value of k_{r_p} increases and both of them fluctuate within certain range. When k_{r_p} equals to 0.5, ε is 2.398m and c is 22.89%. There is a declining tendency for t and o. When k_{r_p} is more than 0.7, t and o take on slightly inverse increasing.

As k_{r_p} increases, the computed virtual force increases, which is benefit to distribution of actors. So t and o are reduced. However, fixed F_{th} and constrains the moving of actors, so t and o will not shrink all the way back.

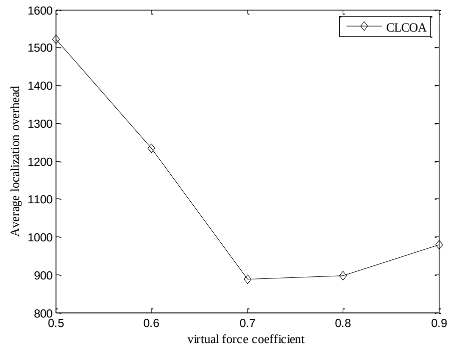

Figure 22. Impact of k_{r_p} on average localization overhead

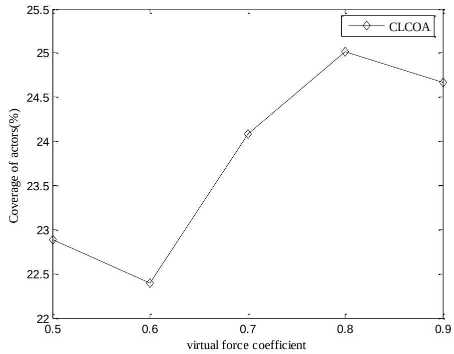

Figure 23. Impact of k_{r_p} on coverage of actors

TABLE I. PERFORMANCE EVALUATION INDEX BEFORE VIRTUAL FORCE MODEL IS INTRODUCED

localization error	localization time	localization overhead	coverage of actors
1.720m	377s	478	17.95%
0.566m	628s	703	11.59%
0.640m	110s	240	11.53%
1.204m	260s	407	13.27%
3.314m	635s	741	9.56%
0.141m	278s	397	11.45%
0.510m	6186s	6299	9.15%
4.205m	408s	513	8.05%
2.433m	693s	800	11.95%
0.849m	1890s	1983	15.22%
0.447m	368s	501	12.91%
3.981m	1210s	2100	11.71%
4.909m	440s	564	15.52%
2.550m	805s	921	17.77%
6.004m	118s	218	12.33%
10.842m	705s	762	14.19%
3.114m	63s	178	14.23%
2.195m	1937s	4018	14.42%
4.610m	905s	936	11.24%
3.106m	1332s	1356	11.76%

TABLE II. PERFORMANCE EVALUATION INDEX AFTER VIRTUAL FORCE IS INTRODUCED

localization error	localization time	localization overhead	coverage of actors
2.440m	316s	850	22.90%
0.412m	408s	1114	24.52%
1.676m	418s	1930	22.71%
0.806m	105s	448	25.22%
1.007m	831s	1800	22.80%
0.583m	1265s	2278	27.95%
2.553m	329s	936	25.61%
0.943m	271s	803	25.57%
2.923m	1630s	3116	25.61%
1.131m	307s	1000	24.16%
0.632m	517s	1336	17.80%
0.618m	555s	1244	22.80%
1.246m	109s	536	24.65%
4.642m	303s	765	17.59%
4.070m	89s	454	23.61%
1.600m	464s	1136	23.36%
1.000m	233s	611	18.43%
6.080m	669s	1558	23.12%
2.518m	2038s	3910	24.62%
5.073m	470s	1127	23.42%

TABLE III. PERFORMANCE COMPARISON BEFORE AND AFTER VIRTUAL FORCE MODEL IS INTRODUCED

index	before	after
ε	2.867 m	2.097 m
t	967s	566 s
o	1205	1347
c	12.79%	23.32%

C. Simulation Experiment 2

Experiment 2 compares the performance before and after virtual force model is introduced. Results of 20 times experiments under the two conditions are provided in both Table I and Table II. And statistic data calculated on this basis are provided in Table III. Table I and Table II show that variation of localization error obviously decreases after virtual force is introduced, however, maximum error reaches 10.842m without virtual force.

It can be observed from Table III that after virtual force is introduced, ε decreases from 2.867m to 2.097m, and t decreases from 967s to 566s. o increases slightly. c obviously increases and is about twice before virtual force is introduced.

Since virtual force is introduced, free actors compute virtual force and adjust their positions after each iteration. It makes all actors well distributed and improves actors' coverage. Meanwhile, the decrease of distance between actors effectively reduces errors caused by moving, thus decreasing localization error and time. Each actor keeps a neighbor table to compute virtual force, which increases localization overhead.

D. Simulation Experiment 3

Experiment 3 compares the performance of CLCOA with centroid algorithm. Centroid algorithm is a representative of range-free technology. Here, we have made some changes to centroid algorithm to make it adapt to our scenario. Values of parameters in experiments are set as below: N_a=14, R_a=50m.

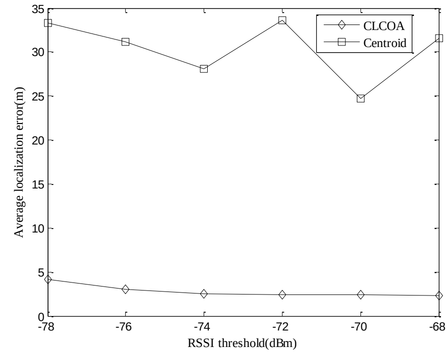

Figure 24. Average localization error

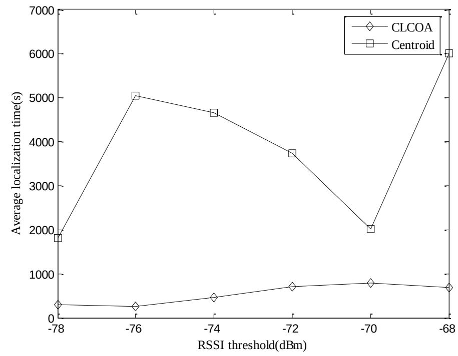

Figure 25. Average localization time

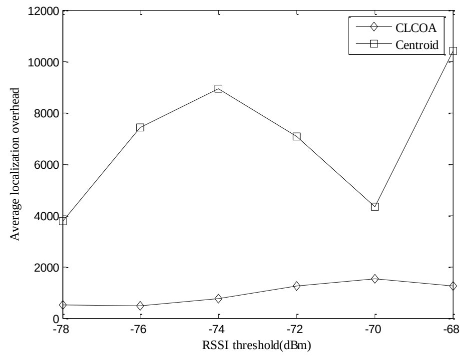

Figure 26. Average localization overhead

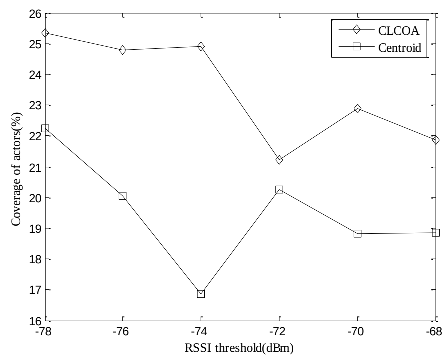

Figure 27. Coverage of actors

Fig. 24-Fig. 27 show comparison of CLCOA and centroid algorithm in ε, t, o and c.

As shown in Fig. 24, ε of CLCOA is much less than that of centroid algorithm. Because in CLCOA the area of the unknown node decreases through iteration, the calculated centroid is more accurate.

It can be observed from Fig. 25 and Fig. 26 that both t and o of centroid algotithm are more than that of CLCOA. When a=-78dBm, the values of t in these two algorithms are 1802s and 292s respectively. In CLCOA, just one actor directly communicates with the unknown node during movement, thus the overhead is less.

It can be obtained from Fig. 27 that coverage of actors in CLCOA is greater compared with centroid algorithm. CLCOA introduces virtual force to make actors well-distributed, so the coverage is optimized.

E. Simulation experiment 4

Experiment 4 compares the performance of CLCOA with the localizing method of single actor. Results of 20 times experiments with the two methods are provided in Table IV-Table VI. Here, only two indexes (localization error and time) are considered. Two group comparison data calculated on this basis are provided in Table VII and Table VIII.

TABLE IV. PERFORMANCE EVALUATION INDEX OF CLCOA

localization error	localization time
2.440m	316s
0.412m	408s
1.676m	418s
0.806m	105s
1.007m	831s
0.583m	1265s
2.553m	329s
0.943m	271s
2.923m	1630s
1.131m	307s
0.632m	517s
0.618m	555s
1.246m	109s
4.642m	303s
4.070m	89s
1.600m	464s
1.000m	233s
6.080m	669s
2.518m	2038s
5.073m	470s

Table V and Table VI show that the localizing method of single actor is very unstable and both localization error and localization time fluctuate greatly. Maximum localization time is 58006s in Table V, whereas the worst case in Table VI is 18.720m.

Statistics on the data in Table IV and Table V are provided in Table VII. It can be observed that under approximate condition of ε, t of CLCOA is 536s, whereas t is 10495s in the localizing method of single actor.

Statistics on the data in Table IV and Table VI are provided in Table VIII. It can be observed that under approximate condition of t, ε of the two methods are 2.009m and 8.711m respectively.

The localizing method of single actor approximately judges distance by RSSI. The signal is unstable and easily affected by the environment. So under approximate condition of t, ε is greater than it in CLCOA. Moreover, the single actor adjusts direction and distance through signal strength to move close to the sensor. So under approximate condition of ε, t of this method is much more than it in CLCOA.

TABLE V. PERFORMANCE EVALUATION INDEX OF THE LOCALIZING METHOD OF SINGLE ACTOR-1 (A=-45DBM)

localization error	localization time
2.061m	3599s
1.414m	3426s
3.780m	50914s
1.749m	11475s
1.082m	181s
2.332m	58006s
1.664m	3792s
1.789m	15576s
2.563m	3505s
1.769m	15440s
2.563m	2410s
2.508m	5823s
2.500m	1246s
1.860m	3140s
3.821m	7436s
1.985m	7619s
3.590m	5457s
1.836m	2640s
1.546m	5111s
1.700m	3120s

TABLE VI. PERFORMANCE EVALUATION INDEX OF THE LOCALIZING METHOD OF SINGLE ACTOR-2 (A=-45DBM)

localization time	localization error
153s	0.906m
167s	7.140m
23s	18.720m
3556s	5.070m
319s	12.985m
344s	10.730m
63s	2.631m
215s	8.832m
2525s	15.232m
44s	12.687m
199s	6.768m
758s	8.223m
324s	6.546m
549s	9.402m
121s	7.034m
386s	12.907m
64s	7.654m
268s	5.941m
106s	5.771m
586s	9.044m

TABLE VII. COMPARISON OF T UNDER APPROXIMATE CONDITION OF E

index	CLCOA	the localizing method of single actor
ε	2.009 m	2.206m
t	536s	10495s

TABLE VIII. COMPARISON OF E UNDER APPROXIMATE CONDITION OF T

index	CLCOA	the localizing method of single actor
t	536s	538s
ε	2.009m	8.711m

F. Summary

In summary, a has the greatest impact on ε, ε decreases as the value of a increases, and other parameters have little impact on ε. Both R_a and N_a have a great impact on c, and the former is more obvious. *c* increases from 22.89% to 63.85% as the value of R_a increases from 50m to 100m. All parameters have impact on *t* and *o* to varying degrees. *t* and *o* increase as the value of *a* and F_{th} respectively increases, whereas *t* and *o* decrease as the value of N_a, R_a and *k*r_p. In addition, the introduced virtual force model makes ε and *t* decrease, but makes *c* and *o* increase.

In addition, compared with the localizing method of single actor and centroid algorithm, performance of CLCOA is better whether in localization error or time.

VI. Conclusion

Based on realizing localization in WSAN, the proposed CLCOA further optimizes distribution of actors in the network through introduced virtual force model. In range-free mode, CLCOA uses mobile actors instead of anchors in WSN to save deployment cost. In addition, compared with complex measurement techniques, CLCOA is simple and requires no additional hardware. The introduction of virtual force makes free actors well distributed while localizing. It guarantees that network events can be disposed in time, thus improving task execution efficiency of the network. Via simulations, it is proven that the performance of CLCOA has improved after virtual force model is introduced, and compared with the localizing method of single actor and centroid algorithm, CLCOA has better locating performance. Our future work is research on range-based localization in WSAN, and combines TOA with time synchronization to realize time-based sensor localization.

Acknowledgment

This work was supported by National Key Technology R&D Program of the Ministry of Science and Technology of China (No.2011BAK21B05), Science and Technology Achievement Transformation Project of National Ministry of Agriculture (No.2011GB2C100001), Science and Technology Support Program of Jiangsu Province (No.BE2011398, No.BE2011339), Fundamental Research Funds for the Central Universities (Y0201100080), Nanjing Agricultural University Youth Science and Technology Innovation Fund (No.Y0201300254), Nanjing Agricultural University National College Students' Innovative Entrepreneurial Training Plan (No. 201310307049).

References

[1] N. Amiya N and S. Ivan S, "Wireless Sensor and Actuator Networks", *Wiley-Interscience, Hoboken*, 2010.
[2] Y. Qiu, C. C. Zhao, G. L. Dai and C. J. Hu, "Research on Localization Technology for Wireless Sensor Networks", *Computer Science*, vol. 35, no. 5, pp. 47-50, 2008.
[3] Y. Peng and D. Wang, "A Review:Wireless Sensor Networks Localization", *Journal of Electronic Measurement and Instrument,* vol. 25, no. 5, pp. 389-396, 2011.
[4] F. B. Wang, L. Shi and F. Y. Ren, "Self-Localization Systems and Algorithms for Wireless Sensor Networks", *Journal of Software,* vol. 16, no. 5, pp. 857-868, 2005.
[5] N. Bulusu, J. Heidemann and D. Estri, "GPS-less low- cost outdoor localization for very small device", *IEEE Person. Commun,* vol. 7, no. 5, pp. 28-34, 2000.
[6] K. J. Mao, X. M. Zhao, W. X. He, M. Xia, L. Wang and Q. Z. Chen, "Area Division Based Semi-auto DV-Hop Localization Algorithm in WSN", *Computer Science,* vol. 39, no. 3, pp. 39-42, 2012.
[7] P. Han, C. S. Gao, M. Yang, D. L. Mao and B. Yu, "ECLS: An Efficient Cooperative Localization Scheme For Wireless Sensor and Actor Networks", *Proceedings of the fifth International Conference on Computer and Information Technology, Shanghai, China,* 2005, pp. 396-400.
[8] G. Cristina, C. M. Pinotti and S. K. Das, "A Semi-Distributed Localization Protocol for Wireless Sensor and Actor Networks", *8th IEEE International Conference on Pervasive Computing and Communications, Mannheim, Germany, 2010,* pp. 438-443.
[9] M. I. Akbas, M. R. Brust and D. Turgut, "Local Positioning for Environmental Monitoring in Wireless Sensor and Actor Networks", *Proceedings of the 35th Annual IEEE Conference on Local Computer Networks, Denver, USA,* 2010, pp. 806-813.
[10] G. A. Shah and O. B. Akan, "Timing-Based Mobile Sensor Localization in Wireless Sensor and Actor Networks", *Mobile Networks and Applications,* vol. 15, no. 5, pp. 664-679, 2010.
[11] S. Y. Ren, "Dynamic wireless sensor and actor networks event driven localization algorithm", Transducer and Microsystem Technologies, vol. 28, no. 12, pp. 89-92, 2009.
[12] G. G. He, "Research on Localization Techniques based on Semidefinite Programming in Wireless Sensor and Actuator Networks", *Southwest Jiaotong University,* 2012.
[13] M. Hussain and N. Trigoni, "Dynamic node placement for multi-hop localization in cluttered environments", *8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, Salt Lake City, USA,* 2011, pp. 167-169.
[14] Z. W. Yuan and J. J. Liang, "Multi-hops localization algorithm based on virtual force for wireless sensor networks", *Journal of Chongqing University of Posts and Telecommunications,* vol. 22, no. 1, pp. 122-126, 2010.
[15] X. P. Ren and Z. X. Cai, "A Distributed Actor Deployment Algorithm for Maximum Connected Coverage in WSAN", *5th International Conference on Natural Computation, Tianjin, China,* 2009, pp. 283-287.
[16] X. Li, "Research on Underground Personnel Localization Technology for Wireless Sensor Networks", *Taiyuan University of Technology,* 2011.
[17] D. K. Z, "Research on RSSI-based Ranging and Localization Technique for WSN", *National University of Defense Technolog, 2010.*
[18] X. Li, "The Localization in Wireless Sensor Networks", *University of Science and Technology of China,* 2009.
[19] R. Lv and X. H. Yang, "Reduction of Wireless Sensor Node Localization Errors", *Tsinghua Science and Technology,* vol. 48, no. S2, pp. 1839-1843, 2008.

Huanliang Xu received his Ph.D. degree from Nanjing University of Aeronautics and Astronautics. He is currently a professor and the vice-president of College of Information Science and Technology at Nanjing Agriculture University. He is also the senior member of China Computer Federation, the vice-chairman of CG&CAD Committee of Jiangsu Computer

Federation, and the standing member of Computer Application Committee of Jiangsu Computer Federation and the one of Jiangsu IoT Special Committee. His current research interests include IoT of agriculture, agricultural informatization, and farm product traceability system.

Duo Li received B.C. degree in computer science and technology from Nanjing Agricultural University in 2007. Currently, she is a postgraduate in the Department of Information Science and Technology, Nanjing Agricultural University. Her main fields of research are WSN localization and WSAN localization.

Jianhua Che received the Ph.D. degree in computer science and technology from Zhejiang University in 2010, the M.S.E. degree in computer software and theory from Yanshan University in 2005, and the B.S.E. degree in computer science and application from Agricultural University of Hebei in 2002, respectively. He is currently working in the College of Information Science and Technology, Nanjing Agricultural University. His current research interests are in the area of Cloud computing, Virtualization system and Big Data.

Haoyun Wang received the Ph.D. degree in information network from Nanjing University of Posts and Telecommunications in 2010. He also received the M.E. degree in communication and information systems from Kunming University of Sciences and Technology in 2005 and the B.E. degree in communication engineering from Huazhong University of Science and Technology in 2002, respectively. Currently, he is working at the College of Information Science and Technology in Nanjing Agricultural University. His current research interests are in the area of WSN, WSAN and CPS.