ISSN 1674-1056
http://cpb.iphy.ac.cn
http://iopscience.iop.org/cpb
中国物理 B
Chinese Physics
Chinese Physics B

A target group tracking algorithm based on a hybrid sensor network

Chun Zhang(张淳)

Citation: Chin. Phys. B . 2018, 27(8): 080101. **doi:** 10.1088/1674-1056/27/8/080101

Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Power control and channel allocation optimization game algorithm with low energy consumption for wireless sensor network

Xiao-Chen Hao(郝晓辰), Jin-Shuo Liu(刘金硕), Li-Xia Xie(解力霞),Bai Chen(陈白), Ning Yao(姚宁)

Chin. Phys. B . 2018, 27(8): 080102. **doi:** 10.1088/1674-1056/27/8/080102

Shannon information capacity of time reversal wideband multiple-input multiple-output system based on correlated statistical channels

Yu Yang(杨瑜), Bing-Zhong Wang(王秉中), Shuai Ding(丁帅)

Chin. Phys. B . 2016, 25(5): 050101. **doi:** 10.1088/1674-1056/25/5/050101

Analysis of robustness of urban bus network

Tao Ren(任涛), Yi-Fan Wang(王一帆), Miao-Miao Liu(刘苗苗), Yan-Jie Xu(徐艳杰)

Chin. Phys. B . 2016, 25(2): 020101. **doi:** 10.1088/1674-1056/25/2/020101

Degree distribution and robustness of cooperativecommunication network with scale-free model

Wang Jian-Rong, Wang Jian-Ping, He Zhen, Xu Hai-Tao

Chin. Phys. B . 2015, 24(6): 060101. **doi:** 10.1088/1674-1056/24/6/060101

A target group tracking algorithm for wireless sensor network using azimuthal angle of arrival information

Zhang Chun, Fei Shu-Min, Zhou Xing-Peng

Chin. Phys. B . 2012, 21(12): 120101. **doi:** 10.1088/1674-1056/21/12/120101

A target group tracking algorithm based on a hybrid sensor network*

Chun Zhang(张淳)†

School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, *China*

(Received 12 November 2017; revised manuscript received 30 April 2018; published online 10 July 2018)

Traditional tracking algorithms based on static sensors have several problems. First, the targets only occur in a part of the interested area; however, a large number of static sensors are distributed in the area to guarantee entire coverage, which leads to wastage of sensor resources. Second, many static sensors have to remain in active mode to track the targets, which causes an increase of energy consumption. To solve these problems, a target group tracking algorithm based on a hybrid sensor network is proposed in this paper, which includes static sensors and mobile sensors. First, an estimation algorithm is proposed to estimate the objective region by static sensors, which work in low-power sensing mode. Second, a movement algorithm based on sliding windows is proposed for mobile sensors to obtain the destinations. Simulation results show that this algorithm can reduce the number of mobile sensors participating in the tracking task and prolong the network lifetime.

Keywords: hybrid sensor network, target group tracking, mobile sensors

PACS: 01.20.+x **DOI:** 10.1088/1674-1056/27/8/080101

1. Introduction

In traditional wireless sensor networks (WSNs), tracking the targets by static sensors is an important and basic application.[1,2] However, there are several problems. First, due to the limited sensing ranges of sensors, a large number of sensors are deployed in an interested area in order to satisfy the coverage property. But during each sampling time, the target group only occurs in a small region of the interested area, hence the deployment of a large number of sensors is a waste of resources.[3–5] Second, the energy of sensors is limited, but the tracking process will consume a great deal of energy. When sensors drain their battery power, many problems will appear in the WSNs, such as communication holes and coverage holes.[6–8] In recent years, with the development of robots and wireless communication, mobile sensors are used in many applications. Compared with static sensors, mobile sensors have some advantages.[9–11] On one hand, mobile sensors can move to any position according to the requirement of target tracking, so the interested area does not need to be completely covered by sensing ranges of sensors. On the other hand, mobile sensors can move close to targets, so the tracking accuracy is improved.

In this paper, we consider the problem of tracking a target group by a hybrid network including mobile sensors and static sensors. However, there are several challenges in order to solve the problem. First, the problem of tracking a target group is different from tracking multiple targets. A group of targets forming a compact body, named a target group, move together from one location to another for a common goal. The problem of tracking multiple targets focuses on tracking the trajectory of each target. A target group is defined as a lot of targets with related motions including associated velocities, directions, accelerations, etc. Since the number of targets within a target group is large and they are near to each other, it is not necessary and it is difficult to track each target. In this paper, the problem of tracking a target group is modeled on tracking a continuous region covered by the target group. Second, it is worth noting that, to reduce the waste of resources, the number of sensors is limited. Hence, the union of sensing ranges of all the static sensors cannot guarantee full coverage of the interested area. But the static sensors are connected with each other. Generally, sensors have three working modes: sleeping mode, sensing mode, and active mode. When a sensor works in active mode, the energy consumed is the largest among the three modes. To save the energy of static sensors, they only work in low-power sensing mode. Under this mode, static sensors periodically sample the movement of the target group, and they only generate the simplest information, which is a binary digit "1". Hence, the first problem is how to schedule the minimum number of mobile sensors during each sampling period to guarantee coverage for the region where the target group appears. Therefore, the objective region should be estimated by deficient static sensor resources. The second problem is, during each sampling period, how to assign the locations for the selected mobile sensors as their destinations. The objective is to maximize the average remaining energy of mobile sensors.

The main contribution of this paper is summarized as follows.

(i) In order to reduce the number of sensors participating in the tracking task, the target group is tracked by several mo-

*Project supported by the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20140875), the Scientific Research Foundation of Nanjing University of Posts and Telecommunications, China (Grant No. NY213084), and the National Natural Science Foundation of China (Grant No. 61502243).

†Corresponding author. E-mail: zhc1088@njupt.edu.cn

© 2018 Chinese Physical Society and IOP Publishing Ltd

bile sensors cooperating with sparsely distributed static sensors. First, an estimation algorithm is proposed. At the beginning of each sampling period, the region where the target group appears is estimated by static sensors with insufficient information. Then, to satisfy the coverage property of the estimated region, the locations where mobile sensors should move to is decided.

(ii) We propose a distributed scheduling algorithm based on sliding windows. According to the predefined weigh function, the movement priority for each mobile sensor is computed. Then, based on the former deployment result for the estimated region, each selected mobile sensor chooses a suitable location as its destination, so that the movement cost satisfies the predefined objective function.

To summarize, static sensors and mobile sensors are assigned to different tasks. Static sensors are responsible for estimating the region where the target group appears. Mobile sensors are responsible for tracking the target group. When mobile sensors sense the targets, they will generate data packets, and send them to the sink. The data packets generated by mobile sensors contain much more information than those of static sensors.

The rest of this paper is organized as follows. Section 2 states the related works. Section 3 and Section 4 define the target group tracking problem and propose our solutions to the problem. Simulation results and conclusions are given in Section 5 and Section 6, respectively.

2. Related works

Wireless sensor networks are used in many application areas. One application is target tracking, which is used in wildlife surveillance, military affairs, environmental monitoring, and so on. Traditional tracking algorithms are based on static sensors.[12] Since most sensors should keep in active mode, sensors consume a great deal of energy to track the target and communicate with other sensors.[13] Hence energy consumption is the main challenge for static sensor networks.[14,15] In order to solve this problem, prediction algorithms have been proposed in many studies.[16–18] In Ref. [19], the location of the target in the next time interval is predicted according to the factors including the previous location, the current acceleration, and speed of the target. Based on the prediction result, fewer sensors are activated to track the target. In Ref. [20], sequential Monte Carlo is used to predict the target position. Then, a dynamic collaboration scheme is proposed for camera tracking. The reference [21] focuses on the target group problem based on the static sensors. First, the definition of a target group is proposed, and the tracking problem is modeled as tracking a monitoring region. Then, a Hull–Algorithm and Cir–Algorithm are proposed to estimate the monitoring region based on binary sensors; however, the sensors cannot move close to the targets, thus the tracking solution is less effective than that of mobile sensors.

In static sensor networks, when the energy of the sensors is exhausted, many problems will exist. Hence, in recent years, many researches introduce the mobile sensors. The reference [22] focuses on the target tracking problem based on heterogeneous camera sensor networks, which consist of a lot of static sensors and a small number of mobile camera sensors. Camera sensors are controlled to track the target in a cascading scheme. In the context of target tracking under various scenarios, the authors in Ref. [23] proposed a tracking solution based on a hybrid network, which consists of a few mobile sensors and a large number of fixed sensors. First, fixed sensors detect the appearance of the target and obtain the current location of the target. Second, mobile sensors close to the target are informed to move toward the target. Hence, the mobile sensors are scheduled to track the target coordinating with fixed sensors. Then, the solution is extended for tracking multiple targets. Based on the action force algorithm, mobile sensors are dispatched to track every single target. In this target tracking scheme, static sensors also need to track the targets. Since the energy consumption of tracking targets is larger than that of sensing targets, static sensors consume a great deal of energy during the tracking process. In order to maximize the lifetime of a network, the authors in Ref. [24] computed the sensing radius, communication radius, and locations of sensors, which cooperate to track a target. However, the energy model does not conclude the movement energy of sensors. In Ref. [25], the authors were concerned with the energy-efficient tracking problem in the presence of obstacles. The field is divided into a grid of small cells. The objective is to find the corresponding locations and shortest path for sensors. In order to solve this problem, the grid is converted to a graph, whose edges are weighted according to the energy consumption of sensors. The energy model consists of sensing, movement, and communication energy. Due to the shortcomings of the traditional particle filter algorithm, an improved particle filter algorithm is designed in Ref. [26] for target tracking. First, the location model is constructed with the features of the target as the measurements. Second, the weight of the particle filter algorithm is optimized. Then, the improved particle filter algorithm is adjusted according to the difference between the current and predicted values. The mobile sensors in Ref. [27] are dispatched to track the target based on the estimation result. First, the location of the target is predicted by the online estimation. Second, the assignment problem of the mobile sensors is modeled as an integer linear programming problem. In Ref. [28], the location of the target is estimated by the improved Kalman filter, and a maximum likelihood is used to deal with the sensing nonlinearity. In sum, many existing references have studied the problem of tracking one or multiple targets; however,

the number of targets within a target group is large and they are very close to each other, thus many existing tracking algorithms are not suitable for the scenario of tracking a target group.

3. A target group tracking algorithm based on a hybrid sensor network

3.1. Problem definitions

Given an interested area G, the target group moves randomly in G. Static sensors and mobile sensors are deployed randomly in G. Initially, we assume they are connected. The number of static sensors is larger than that of mobile sensors. At the beginning of each sampling period, the estimated region of the target group is obtained based on the information generated by static sensors, which is defined as I. In order to guarantee the tracking accuracy, mobile sensors should ensure the complete coverage of I. We assume the union of sensing ranges of mobile sensors is B. The relationship of I and B should satisfy $I \subseteq B$.

Our objective is to ensure both coverage (in the sense that I and B satisfy $I \subseteq B$) and connectivity (in the sense that selected mobile sensors are connected) requirements by scheduling as few mobile sensors as possible. Meanwhile, the movement cost of mobile sensors satisfies the predefined objective function.

3.2. Estimating the objective region

The target group is a group of targets that form a compact individual. Each target within the target group moves around a logical center.[21] The logical center is not a real point in real life. It is a logical point with which the movement of targets are correlated. When the logical center moves from one location to another, the targets move according to the new location of the logical center. In this paper, the distance between a target and the logical center is less than d_{m}. The distance d_{m} is expressed as Eq. (1), where k is the number of targets, and γ is the coefficient. The problem of tracking a target group is modeled as tracking a region covered by the target group, which is defined as the objective region.

$$d_{\mathrm{m}} = \gamma k. \tag{1}$$

As illustrated in Fig. 1, if the motion of the targets is not related, the problem of tracking these targets focuses on tracking the trajectory of each target. However, if the motion of the targets is related to the logical point, the shape of the area covered by the targets changes at different times, but the distances between the targets and the logical center are all less than d_{m}. The problem of tracking these targets focuses on tracking the geographically continuous area covered by them during each sampling period. The border of the area is marked with a dotted line in Fig. 1.

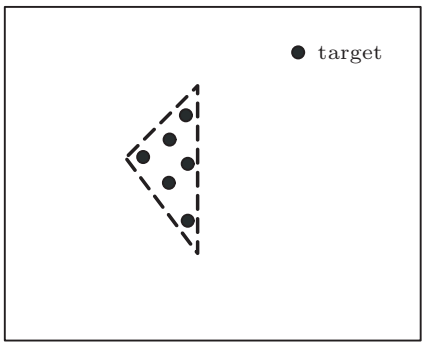


Fig. 1. Illustration of the target group.

Compared with tracking an individual target, the effect of coverage holes is less when tracking a target group. When an individual target moves in a coverage hole, it will be lost. However, when a coverage hole is not large enough, some members within the target group can also be sensed by sensors. Hence, in order to decrease the number of sensors, the union of sensing ranges of all the static sensors does not guarantee full coverage of G; however, the density of static sensors impacts the estimating accuracy.

An estimation algorithm of the objective region is proposed in this section. We assume each static sensor knows the location of itself. At the beginning of each sampling period, if a static sensor senses the targets, it generates a data packet containing a binary digit "1" and its location. Then it sends the data packet to the sink. The set of static sensors which have sensed targets is expressed as $\boldsymbol{U}$.

Since the distance between any target and the logic center is less than d_{m}, the objective region is estimated as a circle. The radius of the estimated objective region is d_{m}. The center of the objective region is approximated by the center of gravity of a quadrilateral. The four vertices of the quadrilateral are the positions of four static sensors selected from $\boldsymbol{U}$. The four vertices are denoted as a, b, c, and d, respectively. The coordinates of them are (x_3, y_3), (x_2, y_2), (x_1, y_1), and (x_4, y_4), respectively. The method for selecting the four sensors is as follows. The horizontal coordinates of sensors in $\boldsymbol{U}$ form a set $\boldsymbol{X}$. The vertical coordinates of sensors in $\boldsymbol{U}$ form a set $\boldsymbol{Y}$. The sink finds the minimum and maximum values in $\boldsymbol{X}$, and they are x_4 and x_2, respectively. The vertical coordinates corresponding to x_4 and x_2 are y_4 and y_2, respectively. Then, the sink finds the minimum and maximum values in $\boldsymbol{Y}$, and they are y_1 and y_3, respectively. The horizontal coordinates corresponding to y_1 and y_3 are x_1 and x_3, respectively. In order to obtain the center of gravity of the quadrilateral, a theorem is given as follows.

Theorem 1 Given a quadrilateral illustrated as shown in Fig. 2, the center of gravity of it is denoted as o. The coordinate of o is expressed as shown in Fig. 2, where a, b, c, and d are the four vertices of the quadrilateral, (x_3, y_3), (x_2, y_2), (x_1, y_1), and (x_4, y_4) are the coordinates of the four vertices.

$$\begin{cases} x_o = \dfrac{\sum\limits_{i=1}^{3} x_i^2 y_{i+1} + \sum\limits_{i=1}^{3} x_i x_{i+1} y_{i+1} - \sum\limits_{i=1}^{3} x_{i+1}^2 y_i - \sum\limits_{i=1}^{3} x_i y_{i+1} y_i + x_4^2 y_1 - x_1^2 y_4 + x_4 x_1 y_1 - x_4 y_1 y_4}{3\left(\sum\limits_{i=1}^{3} x_i y_{i+1} - \sum\limits_{i=1}^{3} x_{i+1} y_i + x_4 y_1 - x_1 y_4\right)}, \\ y_o = \dfrac{\sum\limits_{i=1}^{3} x_i y_{i+1}^2 + \sum\limits_{i=1}^{3} x_i y_i y_{i+1} - \sum\limits_{i=1}^{3} x_{i+1} y_i^2 - \sum\limits_{i=1}^{3} x_i x_{i+1} y_i + x_4 y_1^2 - x_1 y_4^2 + x_4 y_4 y_1 - x_4 x_1 y_4}{3\left(\sum\limits_{i=1}^{3} x_i y_{i+1} - \sum\limits_{i=1}^{3} x_{i+1} y_i + x_4 y_1 - x_1 y_4\right)}. \end{cases} \tag{2}$$

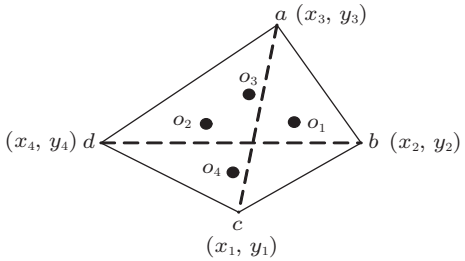


Fig. 2. Illustration of the quadrilateral.

Proof The quadrilateral can be divided into two triangles Δabc and Δadc. The centers of Δabc and Δadc are o_1 and o_2, respectively. The coordinates of them are expressed as Eq. (3). The quadrilateral can also be divided into two triangles Δadb and Δbdc. The centers of Δadb and Δbdc are o_3 and o_4, respectively. The coordinates of them are expressed as Eq. (4). The equations of lines o_1o_2 and o_3o_4 are expressed as Eqs. (5) and (6), respectively.

$$\begin{cases} o_1 \left(\dfrac{x_3+x_2+x_1}{3}, \dfrac{y_3+y_2+y_1}{3}\right), \\ o_2 \left(\dfrac{x_3+x_4+x_1}{3}, \dfrac{y_3+y_4+y_1}{3}\right), \end{cases} \tag{3}$$

$$\begin{cases} o_3 \left(\dfrac{x_3+x_2+x_4}{3}, \dfrac{y_3+y_2+y_4}{3}\right), \\ o_4 \left(\dfrac{x_2+x_4+x_1}{3}, \dfrac{y_2+y_4+y_1}{3}\right). \end{cases} \tag{4}$$

$$\begin{aligned} & y(x_2-x_4) \\ = \; & x(y_2-y_4) + (y_3x_2 + y_1x_2 - y_3x_4 - y_2x_4 \\ & - y_1x_4 - x_3y_2 - x_1y_2 + x_3y_4 + x_2y_4 + x_1y_4)/3, \end{aligned} \tag{5}$$

$$\begin{aligned} & y(x_1-x_3) \\ = \; & x(y_1-y_3) + (y_3x_1 + y_2x_1 + y_4x_1 - y_2x_3 \\ & - y_4x_3 - x_3y_1 - x_2y_1 - x_4y_1 + x_4y_3 + x_2y_3)/3. \end{aligned} \tag{6}$$

The intersection of two lines o_1o_2 and o_3o_4 is the point o. According to Eqs. (5) and (6), we can obtain the coordinate of o, which is expressed as Eq. (2). □

To sum, the coordinate of the center of the estimated objective region is expressed as Eq. (2), and the radius of the estimated objective region is d_m.

Theorem 2 The time complexity of the estimation algorithm is $O(n)$, where n is the number of the elements in $\boldsymbol{U}$.

Proof The center of the estimated objective region is approximated by the center of gravity of a quadrilateral. The center of gravity of a quadrilateral is obtained based on the coordinates of four sensors. The four sensors are selected by traversing all the elements in $\boldsymbol{U}$. The number of elements in $\boldsymbol{U}$ is n. Therefore, it takes $O(n)$ times for the sink finding the four sensors. The total time complexity of the estimation algorithm is $O(n)$. □

After the estimated objective region is obtained, mobile sensors will move to the region to track the target group. Mobile sensors only need to guarantee the coverage of the objective region I. Compared with full coverage of G, the number of mobile sensors moving to cover I is less. Thus, the number of sensors participating in the tracking task is decreased.

According to the deployment scheme, the deployment of mobile sensors can be obtained. Since many deployment schemes have been proposed in many related works, the deployment scheme is not our research focus. The deployment scheme proposed in Ref. [29] is used in this paper. The sensing radius and communication radius of a sensor are expressed as r_s and r_c, respectively. Based on the relationship between r_s and r_c, the deployment solution is separated into two cases. Figure 3 illustrates the deployment solution for mobile sensors, where the dotted circle represents the estimated objective region.

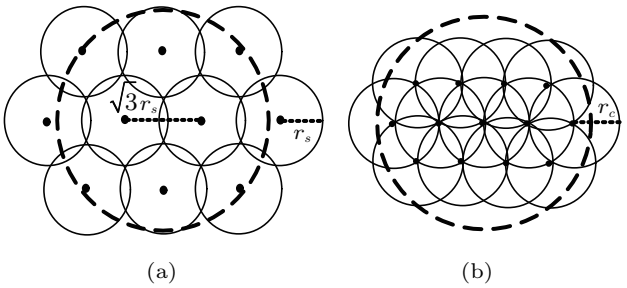


Fig. 3. Illustration of deployment solution for mobile sensors. (a) The deployment solution when $r_c \geq \sqrt{3}r_s$. (b) The deployment solution when $r_c < \sqrt{3}r_s$.

Case 1 $r_c \geq \sqrt{3}r_s$. In this case, the deployment of mobile sensors is as shown in Fig. 3(a). The distance between two neighbor sensors is $\sqrt{3}r_s$.

Case 2 $r_c < \sqrt{3}r_s$. In this case, the deployment of mobile sensors is as shown in Fig. 3(b). The distance between two neighbor sensors is r_c.

3.3. The movement algorithm of mobile sensors

Based on Subsection 3.2, the sink can obtain the locations to be placed with mobile sensors. Then, the sink sends the information of these locations to the mobile sensors.

In this section, some mobile sensors are selected moving to these locations. In this paper, initially, n mobile sensors are randomly selected from M mobile sensors. According to the proposed movement algorithm, each of them selects a location as its destination. The objective is to maximize the average remaining energy of mobile sensors.

The following assumptions are made:

1) The total number of mobile sensors is M. They are connected with each other and their initial energy is the same.

2) At time T, the sink calculates that there are n locations to be placed with mobile sensors, and then broadcasts the coordinates of these locations to mobile sensors.

The objective function of the movement cost is expressed as Eq. (7), where $\boldsymbol{S}$ is the set of n mobile sensors, which move to n locations, s_i is a mobile sensor with the identification of i, IE_i is the initial energy of s_i, ΔE is the energy consumption when a mobile sensor moves the unit distance, and d_i is the distance between s_i and its destination.

$$\max \frac{\sum_{s_i \in \boldsymbol{S}} (IE_i - E \times d_i)}{n}. \tag{7}$$

3.3.1. Calculating the priority function of mobile sensors

At time T, the set of n locations is expressed as Eq. (8), where p_n is a location with the identification of n. $\boldsymbol{L}$ is the set of locations to be placed with mobile sensors. $\boldsymbol{L}$ is obtained by the following three steps.

Step 1 At the beginning of each sampling period, the objective region of the target group, which is defined as I, is estimated by the method proposed in subsection 3.2, and I is a circle.

Step 2 The deployment solution for mobile sensors within the circle I is illustrated in Fig. 3.

Step 3 In order to cover I, the number and the locations of mobile sensors in I can be obtained according to step 1 and step 2.

Therefore, the number and the values of the elements in $\boldsymbol{L}$ are obtained.

The set of mobile sensors which will move to n locations is expressed as Eq. (9). For any sensor in $\boldsymbol{S}$, it can choose any location in $\boldsymbol{L}$ as its destination. Hence, a measurement is designed to compute the priority for a sensor moving to a location. The priority function is expressed as Eq. (10), where s_i is a random sensor in $\boldsymbol{S}$, p_j is an element in $\boldsymbol{L}$, d_{ji} is the distance between s_i and p_j, E_i is the current energy of s_i, α and β are the weights, and β is larger than α,

$$\boldsymbol{L} = \{p_1, p_2, ..., p_n\}, \tag{8}$$

$$\boldsymbol{S} = \{s_1, s_2, ..., s_n\}, \tag{9}$$

$$\sigma_{ji} = \frac{1}{\alpha d_{ji} \beta E_i}. \tag{10}$$

3.3.2. Establishing the priority matrix

Based on the priority function, each sensor in $\boldsymbol{S}$ computes the priority for itself moving to each location in $\boldsymbol{L}$. They exchange the results with each other. Then the priority matrix $\boldsymbol{Z}$ is established and expressed as Eq. (11), where σ_{ji} is the priority for s_i moving to p_j.

$$\boldsymbol{Z} = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{n1} \\ \vdots & \ddots & \vdots \\ \sigma_{1n} & \cdots & \sigma_{nn} \end{bmatrix}. \tag{11}$$

3.3.3. A movement algorithm based on sliding windows

For any sensor in $\boldsymbol{S}$, the location with the highest priority is a priority for its destination. Hence, there exists a problem. Multiple sensors may choose the same location. In order to solve this problem, a movement algorithm based on sliding windows is proposed. This is a process of a bidirectional choice between mobile sensors and locations. Next, we discuss the process of choosing a destination for any sensor s_i in $\boldsymbol{S}$.

Based on $\boldsymbol{Z}$, s_i can obtain a set $\boldsymbol{C}_i$, which is expressed as Eq. (12). Then, sort the elements in $\boldsymbol{C}_i$ in descending order, and a new set $\boldsymbol{A}_i$ is obtained. $\boldsymbol{A}_i$ is expressed as Eq. (13), where a_1 is the element with maximum value in $\boldsymbol{C}_i$, next is a_2, and so on. There is a special case. When the values of some elements in $\boldsymbol{C}_i$ are the same, s_i sorts them in any order. Then put them into a suitable place in $\boldsymbol{A}_i$. A definition of the sliding window is introduced. A set of elements is called a sliding window. The size of a sliding window is the number of the elements. In this paper, the size of a sliding window is 1. Note that, each sensor can only judge whether one location can be its destination at a time. It cannot measure multiple locations at the same time. Hence, the size of a sliding window cannot be larger than 1.

$$\boldsymbol{C}_i = [\ \sigma_{1i}\ \ \sigma_{2i}\ \ \cdots\ \ \sigma_{ni}\ ], \tag{12}$$

$$\boldsymbol{A}_i = [\ a_1\ \ a_2\ \ \cdots\ \ a_n\ ]. \tag{13}$$

Initially, the element in the sliding window is a_1. We assume a_1 is σ_{ei}. Then s_i will choose p_e, and judge whether p_e chooses s_i moving to it. If yes, p_e is the destination for s_i. Otherwise, the sliding window slides a grid to the right. Then the element in the sliding window is a_2. We assume a_2 is σ_{fi}. s_i will repeat the above steps until it obtains its destination. This process is illustrated in Figs. 4(a) and 4(b).

Next we discuss that, when a sensor in $\boldsymbol{S}$ chooses a location in $\boldsymbol{L}$, whether the location chooses the sensor moving to it. It is worth noting that, this process is completed by the sensor.

We assume p_j is a random element in $\boldsymbol{L}$, and s_i chooses p_j. The method for s_i to judge whether p_j chooses s_i is proposed as follows.

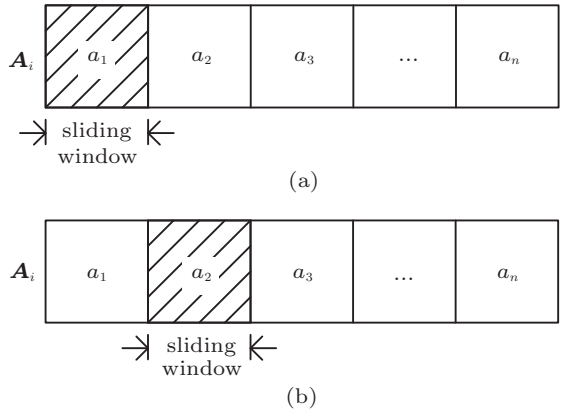


Fig. 4. Sliding window of $\boldsymbol{A}_i$. (a) The sliding window at a_1 and (b) the sliding window at a_2.

Based on $\boldsymbol{Z}$, s_i can obtain a set $\boldsymbol{D}_j$, which is expressed as Eq. (14). Sort the elements in $\boldsymbol{D}_j$ in descending order, and a new set $\boldsymbol{B}_j$ is obtained. $\boldsymbol{B}_j$ is expressed as Eq. (15). There is a special case. When the values of some elements in $\boldsymbol{D}_j$ are the same, s_i sorts them in any order. Then put them into a suitable place in $\boldsymbol{B}_j$. A sliding window is introduced. Initially, the element in the sliding window is b_1. We assume b_1 is σ_{jg}. The discussion is separated into two cases.

$$\boldsymbol{D}_j = [\sigma_{j1} \ \ \sigma_{j2} \ \ \cdots \ \ \sigma_{jn}], \tag{14}$$

$$\boldsymbol{B}_j = [\ b_1 \ \ b_2 \ \ \cdots \ \ b_n]. \tag{15}$$

Case 1 $g = i$. In this case, p_j chooses s_i. Hence, p_j is the destination of s_i.

Case 2 $g \neq i$. In this case, there are two situations.

1) If p_j is the highest priority for s_i, p_j does not choose s_i, and s_i marks s_g as occupied. It is worth noting that, if a sensor is marked as occupied, it is the preferred choice for a location with higher priority. Then the locations with lower priority will not choose it.

2) If p_j is not the highest priority for s_i, s_i will check whether s_g is marked as occupied. If yes, s_g has been chosen by another location with higher priority than p_j, and p_j cannot choose s_g. Then the sliding window slides a grid to the right. We assume the current element in the sliding window is σ_{jh}. According to the relationship between h and i, s_i will repeat the steps in Case 1 or Case 2 until it obtains its destination. Else if s_g is not marked as occupied, p_j do not choose s_i and s_i marks s_g as occupied.

To better explain the process, we present an example scenario depicted in Fig. 4 and Fig. 5. We assume a_2 is σ_{ji} in Fig. 4(b). When s_i judges whether p_j chooses s_i, the movement of the sliding window is illustrated in Figs. 5(a) and 5(b). Figure 5(a) shows the set of elements in $\boldsymbol{B}_j$. Initially, the element in the sliding window is b_1. According to Fig. 4(b), p_j is not the highest priority in $\boldsymbol{A}_i$. We assume that b_1 is σ_{jg} and s_g is marked as occupied, where s_g satisfies the inequality $g \neq i$. Then, as shown in Fig. 5(b), the sliding window slides a grid to b_2. We assume b_2 is σ_{jh}. If s_h satisfies $h = i$, p_j is the destination of s_i. Otherwise, there are two situations.

1) If s_h has not been occupied by s_i, p_j does not choose s_i, and s_i marks s_h as occupied.

2) If s_h has been occupied, the sliding window slides a grid to b_3. Then repeat the above process.

After performing the algorithm, there is no possibility that one mobile sensor in $\boldsymbol{S}$ selects multiple locations. In addition, the number of elements in $\boldsymbol{S}$ is the same as that in $\boldsymbol{L}$. Therefore, there is no possibility that some mobile sensors do not choose any location.

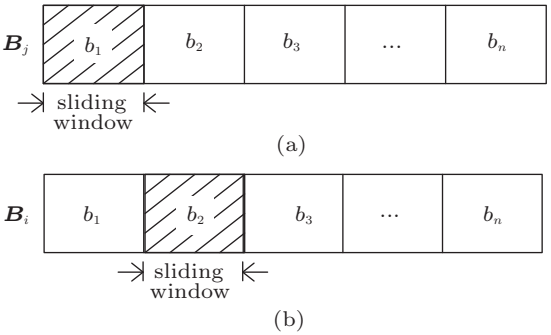


Fig. 5. Sliding window of $\boldsymbol{B}_j$. (a) The sliding window at b_1 and (b) the sliding window at b_2.

The flow chart for any sensor s_i in $\boldsymbol{S}$ choosing a location as its destination is illustrated in Fig. 6.

3.3.4. Confirming the destinations of sensors

After performing the algorithm in Subsubsection 3.3.3, most mobile sensors in $\boldsymbol{S}$ have chosen different locations in $\boldsymbol{L}$; however, there is a possibility that a small number of mobile sensors still choose the same location. The reason is that, in order to reduce the energy consumption, mobile sensors do not exchange information with each other during the selection process. Hence, it is possible that, when a mobile sensor judges that a location is not selected by others, it is actually selected by others. In this circumstance, some locations in $\boldsymbol{L}$ are not chosen by any sensor. In order to solve this problem, a confirmation method is proposed in this section.

Three definitions are given as follows.

$\boldsymbol{Q}$: The set of locations which have not been chosen by any sensor in $\boldsymbol{S}$ as destinations, and satisfies the relationship $\boldsymbol{Q} \subset \boldsymbol{L}$.

$\boldsymbol{F}$: The set $\boldsymbol{F}$ satisfies the relationship $\boldsymbol{F} \subset \boldsymbol{L}$. Each location in $\boldsymbol{F}$ is chosen by multiple sensors in $\boldsymbol{S}$ as their destinations.

$\boldsymbol{T}$: The set $\boldsymbol{T}$ satisfies the relationship $\boldsymbol{T} \subset \boldsymbol{S}$ and is expressed as Eq. (16). $\boldsymbol{T}'(s_k)$ is the set of sensors which choose the same location p_k as their destination.

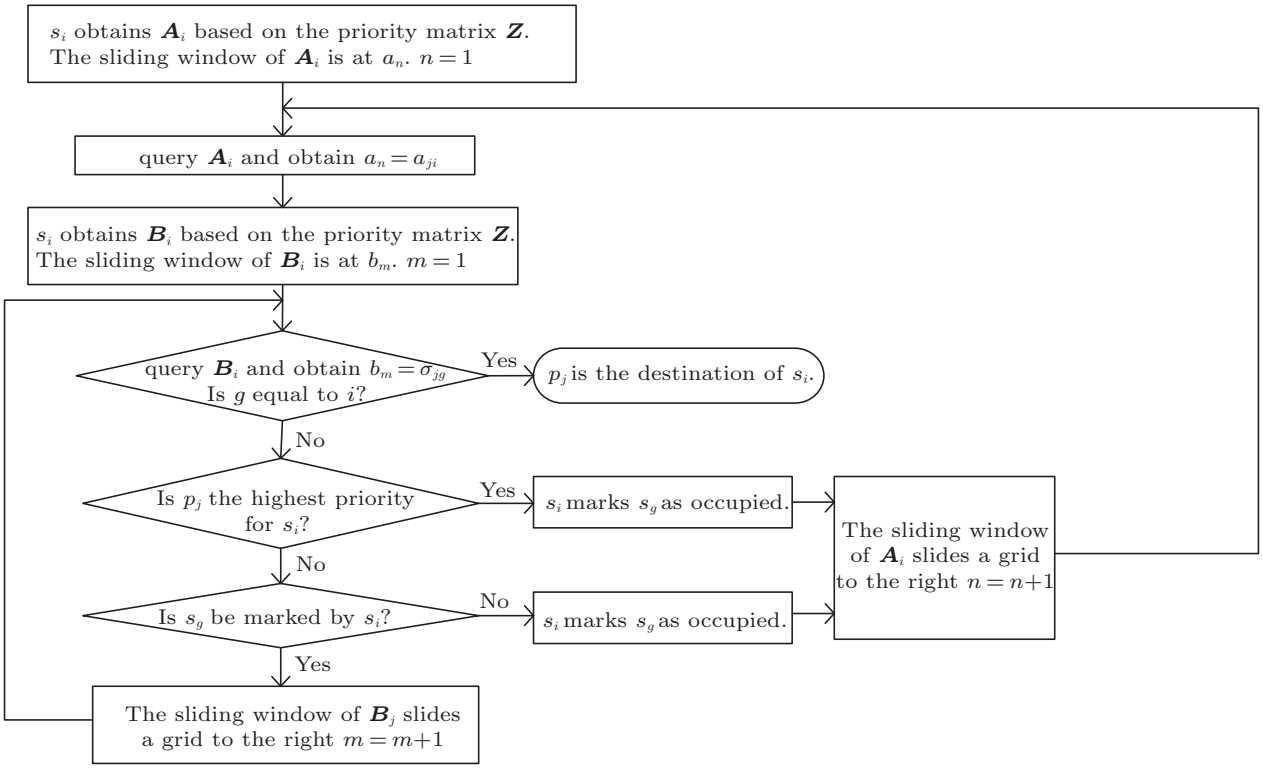


Fig. 6. The flow chart for s_i choosing destination.

Each sensor in $\boldsymbol{S}$ obtains its destination by performing the algorithm in Subsubsection 3.3.3. They exchange the results with each other. If a sensor finds that its destination is the same as another sensor, it constructs the sets $\boldsymbol{T}$, $\boldsymbol{F}$, and $\boldsymbol{Q}$. For each location p_k in $\boldsymbol{F}$, the sensor in $\boldsymbol{T}'(p_k)$ with the largest identification will move to p_k. The other sensors in $\boldsymbol{T}'(p_k)$ cannot move to p_k. Then, according to the algorithm in Subsubsection 3.3.3, each of them chooses a location in $\boldsymbol{Q}$ as its destination. Repeat the above steps until equation (17) is satisfied.

$$\boldsymbol{T}=\bigcup_{p_k\in\boldsymbol{F}}\boldsymbol{T}'(p_k), \tag{16}$$

$$\boldsymbol{Q}=\phi. \tag{17}$$

4. Discussion

In this section, some theorems are given as follows.

Theorem 3 The estimation error of the target group is less than the value of $r_{\rm s}+d_{\rm m}$, where $r_{\rm s}$ is the sensing range of a static sensor, and $d_{\rm m}$ is the maximum distance between a target and the logical center of the target group.

Proof The estimation error is defined as the distance between the center of the estimated objective region and the logical center of the target group. The distance between a target and a static sensor which has sensed a target is less than $r_{\rm s}$. The distance between a target and the logical center of the target group is less than $d_{\rm m}$. Thus, the distance between a static sensor and the logical center is less than $r_{\rm s}+d_{\rm m}$. Then, we can obtain that, the distances between the logical center and the four vertices of the quadrilateral $abcd$ are less than $r_{\rm s}+d_{\rm m}$. The center of the estimated objective region is the center of gravity of the quadrilateral $abcd$. Therefore, the estimation error is less than $r_{\rm s}+d_{\rm m}$. □

Theorem 4 The time complexity of the movement algorithm based on sliding windows is $O(n^2)$, where n is the number of locations in $\boldsymbol{L}$.

Proof As illustrated in Figs. 4(a) and 4(b), a sensor S_i searches for its destination according to A_i. In the worst case, S_i finds its destination until the sliding window slides to the last grid a_n. Under this circumstance, the sliding window moves n grids in total. When the sliding window stays in a grid, S_i will judge whether the corresponding location p_j is its destination. This judgment process is according to B_j. In the worst case, S_i confirms whether p_j is its destination until the sliding window slides to the last grid b_n. During this process, the sliding window also moves n grids in total. Therefore, in the worst case, it takes $O(n^2)$ times for S_i finding its destination. The total time complexity of the movement algorithm is $O(n^2)$. □

5. Simulation results

In this section, we carry out extensive simulation tests to evaluate the performance of our algorithm. The target group tracking algorithm proposed in this paper is called HNTA. In our simulations, we run the algorithms 10 times, and use 10

random trajectories for the logical center of the target group. In each test, sensors are randomly deployed. The simulation results are the averages of 10 tests. The parameters for the network setup are shown in Table 1. The unit energy cost of mobile sensors ΔE, which is specified in Ref. [29], is used in this paper.

Table 1. Simulation parameters.

Parameters	Values
Interested area size/m^2	400×400
Number of simulations	10
Velocity of targets/(m/s)	0.3
Velocity of mobile sensors/(m/s)	2
α	1
β	2
γ	2
E/(energy unit per meter)	1
Initial energy of a mobile sensor (energy unit)	1000

Figure 7 focuses on the effect of the number of static sensors on the estimation error. In this experiment, the number of targets is 20. The setting of r_s is 20. We can observe that the estimation error decreases as the number of static sensors increases. With the increase of the number of static sensors, more static sensors have sensed the targets. The sink can obtain more information about the targets. Thus the tracking accuracy is enhanced. We can also find that the estimation error is less than the value of $r_s + d_m$. This result is consistent with Theorem 3.

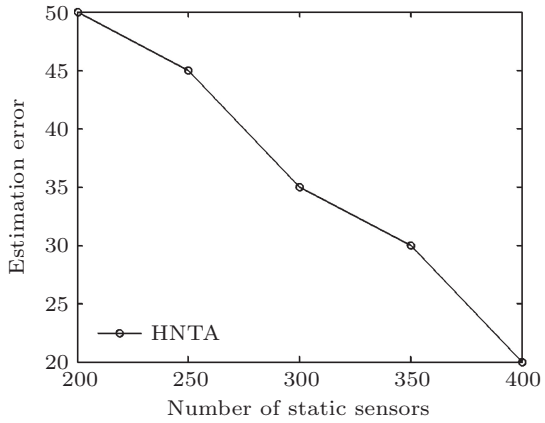


Fig. 7. Number of static sensors versus estimation error.

Figures 8–10 concern the number of sensors tracking the targets. In HNTA, the task of the static sensors is working in the low-power sensing mode to estimate the objective region, while the task of the mobile sensors is tracking the target group. Hence, figures 8–10 evaluate the number of mobile sensors tracking targets in HNTA. We compare our proposed HNTA with the algorithm proposed in Refs. [30] and [31], which are called SOTA and DEETH respectively. In SOTA, the network is organized into several grids. During each sampling period, minimum sensors in a grid are schedule to track the targets.

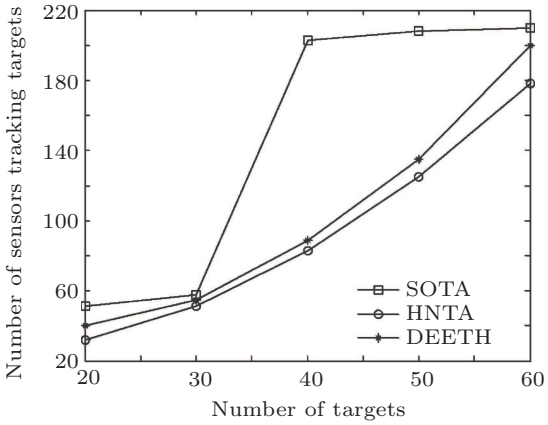


Fig. 8. Number of targets versus number of sensors tracking targets.

Figure 8 shows the average number of sensors participating in the tracking task during each sampling time when the number of targets increases from 20 to 60. In this experiment, the number of static sensors and mobile sensors are 500 and 300, respectively. They are randomly deployed in an interested area. Initially, static sensors and mobile sensors are connected, respectively. We can observe that, in HNTA, when the number of targets within a target group increases, the number of mobile sensors selected by the sink to track the targets increases. The reason is that, with the increase in the number of the targets, the estimated objective region is expanded. More sensors are needed to cover the estimated objective region entirely. We can also find that, in SOTA, the number of sensors scheduled to tracking the targets is larger than that of HNTA. In addition, when further increasing the number of targets, the number of sensors for tracking targets in SOTA is significantly increased. This is because, in SOTA, the interested area is divided into grids. When the targets move in a grid, the minimum number of sensors are selected to cover the grid. However, the targets may only occur in a part of a grid, hence the entire coverage of a grid is a waste of resources. When the number of sensors is not large enough, the targets may only occur in a grid. With the increasing number of the targets, the targets will occur in several grids. The sensors in these grids are scheduled to track the targets. Therefore, the number of sensors tracking the targets increases obviously. The number of mobile sensors searching for targets in DEETH is also evaluated. We can find that the number of mobile sensors tracking targets in DEETH is larger than HNTA. After static sensors are deployed, although static sensors are connected with each other, there may be some coverage holes. In this circumstance, the static sensors may only sense some targets, and the other targets cannot be sensed by static sensors. In DEETH, the mobile sensors need to move

randomly by the swarm technique to search for them. Thus, in order to find these targets quickly, more sensors need to be schedule in DEETH. However, in HNTA, first, according to the information of static sensors which have sensed targets, the region where the target group may appear is estimated. Then, the mobile sensors only need to cover the estimated region. In another circumstance, when all the targets are sensed by static sensors, in DEETH, for each static sensor which have sensed targets, at least one mobile sensor moves towards it. In HNTA, the deployment solution for mobile sensors in the objective region reduces the overlap of sensing ranges. Then the number of mobile sensors tracking the targets is reduced. Thus, HNTA performs better than DEETH.

Figure 9 evaluates the effect of r_s on the number of sensors tracking the targets. In this experiment, the number of targets is 20. To reflect the relationship of $r_c > \sqrt{3}r_s$, the settings of (r_c, r_s) are $(9,5)$, $(12.6,7)$, $(18,10)$, and $(21.6,12)$, respectively. We can find that, with the increase of r_s, the performance achieved by HNTA, DEETH, and SOTA trends downward, and HNTA outperforms DEETH and SOTA. The reason is that, when r_s increases, less sensors are dispatched to cover the region where the targets move in.

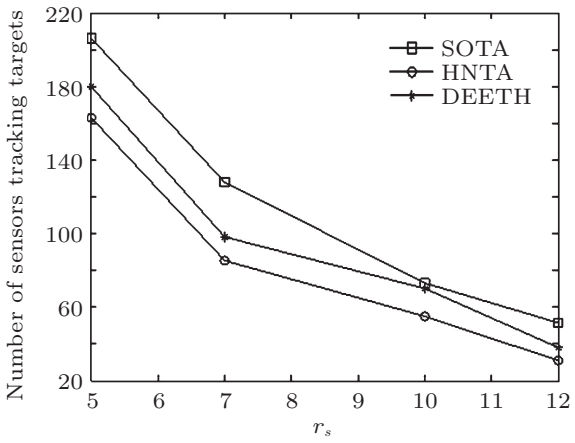


Fig. 9. r_s versus number of sensors tracking targets.

Figure 10 highlights the effect of the ratio r_c/r_s on the number of sensors tracking the targets. In this experiment, to reflect the relationship from $r_c < \sqrt{3}r_s$ to $r_c > \sqrt{3}r_s$, the settings of (r_c, r_s) are $(10,12)$, $(13,12)$, $(19,12)$, $(22,12)$, and $(28,12)$, respectively. The number of targets is 20. Compared with $r_c > \sqrt{3}r_s$, the number of sensors tracking the targets is more when $r_c < \sqrt{3}r_s$. This is because, in HNTA, the deployment scheme of sensors is illustrated by Fig. 3(a) and Fig. 3(b) when $r_c > \sqrt{3}r_s$ and $r_c < \sqrt{3}r_s$, respectively. We can also observe that, HNTA performs better than DEETH and SOTA.

The energy consumption of a sensor consists of three parts: energy consumption in the sleeping state, active state, and sensing state. The energy consumed in the sensing state is smaller than that in the active state. In addition, the energy of static sensors is limited, and they cannot change the battery. Hence, in the algorithm proposed in this paper, static sensors work in sensing state to reduce the energy consumption, but mobile sensors work in active state. Static and mobile sensors work together to track the target group. In the other two algorithms, DEETH and RM, mobile sensors also work in the active state. The focus of the three algorithms is the motion strategy of the mobile sensors, and the energy consumption of mobile sensors is much more than that of static sensors. Therefore, in the simulation experiment, we only evaluate the energy consumption of the mobile sensors in the three algorithms, without considering the energy consumption of the static sensors.

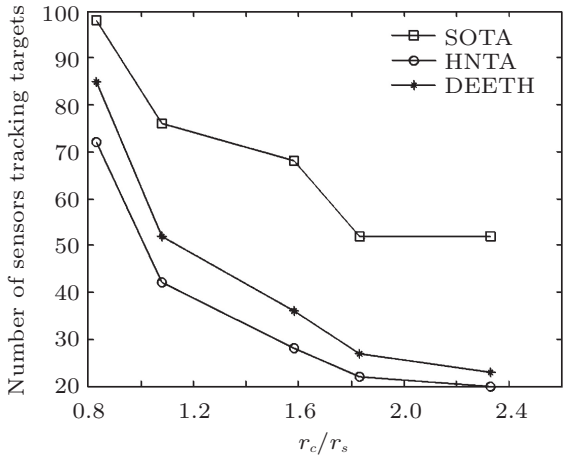


Fig. 10. r_c/r_s versus number of sensors tracking targets.

Figures 11–12 evaluate the effect of movement strategy on the energy consumption of mobile sensors. The object is during each sampling period, the targets are found by mobile sensors. Figure 11 shows the average remaining energy of sensors when the number of targets changes. The setting of (r_c, r_s) is $(20,10)$. We compare our algorithm with DEETH and a random method designed in Ref. [29], which is called RM. The main idea of the RM is, during each sampling period, the sink randomly selects a location in $\boldsymbol{L}$ for each mobile sensor in $\boldsymbol{S}$ as its destination. In Fig. 11, the average remaining energy of sensors achieved by HNTA is higher than that of RM. The reason is that, the process for a sensor to find its destination is a matching problem. In HNTA, this problem is modeled as a bidirectional choice between $\boldsymbol{L}$ and $\boldsymbol{S}$. The matching results are obtained according to the priority function. Thus, compared with RM, the HNTA can prolong the network lifetime. In the tests, we evaluate the energy consumption of the searching process in DEETH. We also assume that, in HNTA and DEETH, when a mobile sensor moves towards a location, it moves along the line between them. We can also find that the remaining energy in HNTA is more than in DEETH. The reason is that, in DEETH, when there are some targets not sensed by static sensors, mobile sensors move randomly by the swarm

technique to find them. In HNTA, although there are coverage holes, the object region of the target group can also be estimated. Hence, in order to find the targets, the moving distance of mobile sensors in HNTA is shorter than DEETH.

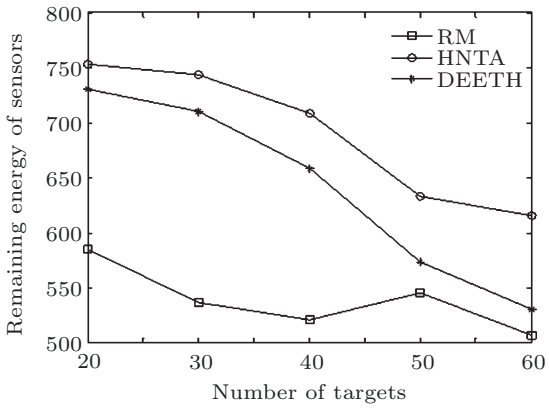


Fig. 11. Number of targets versus remaining energy of sensors.

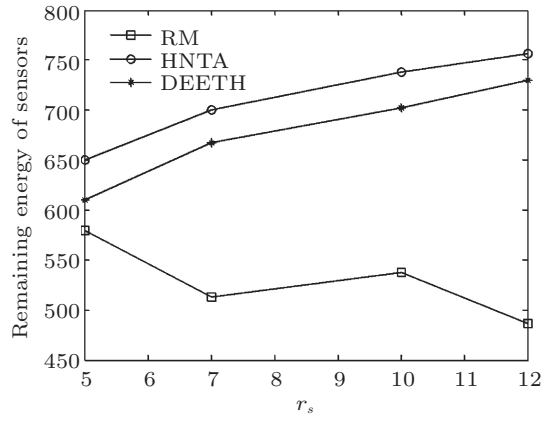


Fig. 12. r_s versus remaining energy of sensors.

Figure 12 shows the average remaining energy of sensors when r_s changes. The number of targets is 20. The setting of r_c is 15. We can observe that, HNTA performs better than RM and DEETH. Therefore, the HNTA can effectively balance the energy consumption of mobile sensors.

6. Conclusions

A target group tracking algorithm based on a hybrid sensor network is proposed in this paper. The objective region is estimated by static sensors, which works in low-power sensing mode. Then, each selected mobile sensor chooses a location as its destination based on the movement algorithm. Algorithm analysis provides the fundamental limits of the estimation algorithm, and the time complexity of the movement algorithm. Simulation results show that this algorithm can decrease the number of mobile sensors tracking the targets, and effectively balance the energy consumption of mobile sensors.

References

[1] Akyildiz I F, Su W, Sankarasubramaniam Y and Cayirci E 2002 *Comput. Networks* **38** 393
[2] Fadel E Gungor V C Nassef L, Akkari N, Maik M G A Almasri S and Akyildiz I F 2015 *Comput Commun.* **71** 22
[3] Chen Z, Cao J C 2013 *Chin. Phys. B* **22** 059201
[4] Wang J R, Wang J P, He Z and Xu H T 2015 *Chin. Phys. B* **24** 060101
[5] Liu H R Dong M R Yin R R and Li H 2015 *Chin. Phys. B* **24** 050506
[6] Zheng X J Cui B T Lou X Y and Zhuang B 2017 *Chin. Phys. B* **26** 040201
[7] Kim K 2013 *Sensors* **13** 11314
[8] Zorbas D and Razafindralambo T 2013 *Comput. Commun.* **36** 1039
[9] Halder S and Ghosal A 2016 *Wireless Networks* **22** 2317
[10] Ren T, Wang Y F, Liu M M and Xu Y J 2016 *Chin. Phys. B* **25** 020101
[11] Yang Y, Ding S and Wang B Z 2016 *Chin. Phys. B* **25** 050101
[12] Mahapatra A, Anand K and Agrand D P 2006 *Comput. Commun.* **29** 437
[13] Atia G K, Veeravalli V V and Fuemmeler J 2011 *IEEE Trans. Signal Process.* **59** 4923
[14] Chen W P, Hou J C and Sha L 2004 *IEEE Trans. Mob. Comput.* **3** 258
[15] Lai Y X Xie J S Lin Z YWang T and LiaoMH 2015 *Sensors* **15** 23218
[16] Samarah S, Al-Hajri M and Boukerche A 2011 *IEEE Trans. Veh. Technol.* **60** 656
[17] Xue L, Liu Z X and Guan X P 2011 *J. Syst. Eng. Electron.* **22** 347
[18] BhuiyanMZ AWang G J Zhang L and Peng Y 2010 *Journal of Central South University of Technology* **17** 340
[19] Rostami A S Mohana F and Keshavarz H 2017 *Wireless Personal Communications* **95** 3585
[20] Zhao S and Yu L 2017 *China Commun.* **14** 44
[21] Cao D L, Jin B H, Das S K and Cao J N 2010 *J. Parallel Distrib. Comput.* **70** 825
[22] Wang T, Peng Z, Xu W Z, Liang J B, Wang G J, Tian H, Cai Y Q and Chen Y H 2017 *Asian J. Control.* **19** 1350
[23] Wang T, Peng Z, Liang J B, Wen S, Bhuiyan M Z A, Cai Y Q and Cao J N 2016 *ACM Trans. Sens. Netw.* **12** 1
[24] Mahboubi H, Momeni A, Aghdam A G, Sayrafian-Pour K and Marbukh V 2017 *IEEE Trans. Control Syst. Technol.* **20** 1522
[25] Mahboubi H, MasoudimansourW, Aghdam A G and Sayrafian-Pour K 2017 *IEEE Trans. Cybern.* **47** 511
[26] Hu F J and Tu C 2017 *J. Intell. Fuzzy Syst.* **32** 3509
[27] Qi Y F, Cheng P, Bai J, Chen J M, Guenard A, Song Y Q and Shi Z G 2016 *IEEE Trans. Ind. Electron.* **63** 6949
[28] Wang X B, Fu M Y and Zhang H S 2012 *IEEE Trans. Mob. Comput.* **11** 567
[29] Wang Y C, Hu C C and Tseng Y C 2008 *IEEE Trans. Mob. Comput.* **7** 262
[30] Khedr A M and Osamy W 2011 *J. Parallel Distrib. Comput.* **71** 1318
[31] Nighot M, Ghatol A and Thakare V 2018 *Int. J. Commun. Syst.* **31** 1