Information Sciences 292 (2015) 95–110

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Probabilistic coverage based sensor scheduling for target tracking sensor networks

Ke Shi *, Hongsheng Chen, Yao Lin

School of Computer Science and Technology, Huazhong University of Science & Technology, Wuhan 430074, China

ARTICLE INFO

Article history:
Received 27 February 2014
Received in revised form 12 August 2014
Accepted 29 August 2014
Available online 6 September 2014

Keywords:
Probabilistic coverage
Sensor scheduling
Target tracking
Wireless sensor networks

ABSTRACT

In target tracking sensor networks, tracking quality and network lifetime are two conflicting optimization goals due to the limited battery power of the sensor nodes. During the movement of a target, how to select an optimal subset of sensors to wake up is of critical importance both for extending network lifetime and guaranteeing tracking quality. In this paper, we first propose a probabilistic-based dynamic non-complete k-coverage method, α–k-coverage, which can guarantee that target moving area is covered by at least k sensors under at least α probability. Then, we propose an energy-efficient sensor scheduling scheme, Optimal Cooperation Scheduling Algorithm (OCSA), to balance tracking quality and network lifetime under α–k-coverage condition. The effectiveness of the proposed scheme is validated through extensive simulation experiments.

1. Introduction

Wireless sensor networks (WSNs) are an emerging technology that has many applications. These networks are composed of hundreds, and potentially thousands of tiny sensor nodes, functioning autonomously, and serving different purposes. They are typically battery powered with limited communication and computation abilities. Each node is equipped with a variety of sensing modalities, such as acoustic, seismic, and infrared. Applications of WSNs include battlefield surveillance, environment monitoring, biological detection, home appliance and inventory tracking. In this paper, we focus on target tracking application.

Target tracking is an important application of wireless sensor networks, such as vehicle tracking in military surveillance [32] and wild animal tracking in habitat monitoring [7,20]. In these applications, tracking quality and network lifetime are two conflicting requirements due to the limited battery power of the sensor nodes. With unlimited power supply, a given area can be monitored perfectly with a set of sensor nodes that cover the entire area in terms of sensing. However, since the sensor nodes have limited power, the quality of monitoring becomes inversely proportional to the life time of the network. Thus, during the movement of a target, how to select an optimal subset of sensors to wake up is of critical importance both for extending network lifetime and guaranteeing tracking quality [11,38]. It is a challenge task to design an energy-efficient tracking algorithm for target tracking application due to the limited battery power of the sensor node, which aims at increasing the network lifetime.

Most works recently in target tracking application focus on the deployment phase of wireless sensor networks. How to schedule sensors dynamically and timely by taking into account of both the energy consumption and tracking quality is still

* Corresponding author.
E-mail addresses: keshi@mail.hust.edu.cn (K. Shi), chenhs1981@163.com (H. Chen), catherinelin0755@qq.com (Y. Lin).

http://dx.doi.org/10.1016/j.ins.2014.08.067
0020-0255/

a challenge during the target's movement. As power is always a valuable and limited resource in sensor networks, it has been advocated that only a small subset of sensor nodes is powered on for the purpose of surveillance and tracking, in which node activation is based on trajectory prediction. However, corresponding performance highly depends on the accuracy of mobility prediction algorithms. In fact, the prediction of the next location of target may not be always accurate. A wrong prediction may cause a wrong sensor scheduling.

Though we cannot predict the next location of target accurately, we can predict the possible moving area of target at next time point, that is, we can predict the approximation bound of the next location of target. In this paper, we propose a dynamic moving area prediction method and a probabilistic-based dynamic non-complete k-coverage method, α–k-coverage, which can guarantee that target moving area is covered by at least k sensors under at least α probability. Thus, a candidate sensor set can be selected using α–k-coverage algorithm. Then, we propose a novel sensor selection algorithm, Optimal Cooperation Scheduling Algorithm (OCSA), to select a suitable sensor subset from the candidate set by both taking into account of minimizing communication cost (minimizing the transmission energy) and sensor's availability. The goal is to find the suitable subset of sensors which will be waked up in next tracking period, in order to not only minimize the transmission energy but also provide certain tracking quality guarantee.

This study is a combination of theoretical analysis and simulated evaluations, the correctness of our tracking algorithm and the effectiveness of the proposed scheduling scheme are validated through theoretical proofs and extensive simulation experiments. The contributions of this paper are the following:

(1) Taking account into tracking quality and network lifetime in target tracking sensor networks.
(2) Proposing the concept of α–k-coverage, a probabilistic-based dynamic non-complete k-coverage method for target tracking application. The candidate sensor set which may be activated at next tracking period can be selected by α–k-coverage.
(3) Designing a novel energy-efficient sensor scheduling algorithm to select a suitable subset from the candidate sensor set.
(4) Analyzing the performance of our approach through simulation.

The rest of the paper is organized as follows. In Section 2, we categorize the related research works in current literature. The system model of target tracking sensor network including the energy consumption model is given the Section 3. The problem definition is also given in Section 3. In Section 4, we propose a probabilistic-based dynamic area coverage algorithm to determine the candidate tracking sensor set. An energy-efficient sensor selection algorithm to balancing the tracking quality and energy consumption by selecting a suitable subset from the candidate sensor set is presented in Section 5. The detailed results of performance evaluation study are presented in Section 6. Finally, we conclude the paper and point out some future works related to this topic.

2. Related works

Tracking moving targets in large scale sensor networks has gained extensive attention recently. Aslam et al. [4] and Mechitov et al. [22] propose several tracking schemes based on the minimalist binary sensor model, in which each sensor's value is converted reliably to one bit of information. This bit indicates that whether the object is moving toward the sensor or away from the sensor. The tracking scheme is then designed based on the area overlapping. However, this binary model cannot measure the distance, and the area overlapping requires large amount of nodes to determine the target's location accurately. It causes heavy energy consumption.

A novel target tracking protocol [31] is proposed by using sensor networks for mobile users. It is assumed that a mobile target may move in any way, so in all the ways the sensor nodes need to be active, thus it consumes too much energy. To save the energy, the number of nodes that actively track the target should be minimized. Most nodes should be in sleeping mode. To guarantee the tracking quality, the nodes around the current location of target should be waked up in time. Information-driven target tracking schemes are proposed by Chu et al. [10] and Zhao et al. [42]. The information utility is computed based on the cost of communication and computation to decide which nodes should actively participate the tracking. The leader is selected to perform this computing. However, it is a one-to-one based handoff scheme. The leader is heavily loaded.

Wang et al. [35], Chen et al. [9], and Yang and Sikdar [37] propose cluster-based tracking schemes. In these schemes, sensor nodes are grouped into clusters either statically or dynamically in the vicinity of target. This cluster is in charge of tracking. The trilateration technique [9], the Voronoi diagram-based approach [35] and KF/MLE based approach [34] are utilized to locate the target. A cluster head coordinated the tracking activities of nodes in this cluster. The key of these schemes is to predict the target's location accurately and construct corresponding cluster based on predicted location.

Zhang and Cao [40,41] introduce a tree-based tracking approach (DCTC). They define a dynamic convoy tree-based collaboration tracking mechanism and formalize the tracking problem as a multiple objective optimization problem. The solution to the problem is a convoy tree sequence with high tree coverage and low energy consumption. However, global network information that is not available in largest sensor network is required for constructing such a convoy tree sequence. Re-configuration and maintenance of a convoy tree incurs considerable computational and communication overhead. And building such a tree also depends on trajectory prediction.

Some other prediction-based tracking algorithms [24,36] are proposed to use prediction method to limit number of sensor nodes to track a target. Specifically, mobile agents are used for tracking. The agents move between sensor nodes with the tracking data [25] indicating the target's moving trace. Marculescu et al. [21] proposed a tracking method to overcome the limitations of continuous location tracking. Sensor nodes operate in a passive way: they only record and spread information about observed target presence in their vicinity. Powerful tracking agent processes this information and locates the target. This agent balances the tracking latency and the energy dissipation. However, the communication overhead is large and powerful agent may not exist in the practical environment. Li et al. [18] developed as a decentralized tracking strategy based on a partial information broadcasting scheme (PIBS), where only a part of the nodes broadcast their tracking estimation results to their neighbors. The finite sensing range of a sensor node is considered and a node activation scheme with variable activation radius is introduced for energy saving.

Liang et al. [19] proposed a distributed infectious disease model (DIDM) for design efficient tracking protocols. The DIDM based wakeup method is derived through establishing the correspondence between sensor wakeup and disease propagation. Besides, one theorem about parameter design is presented, exploiting the relationship among sensor properties, communication properties, performance requirements and the method parameters.

All these schemes are all dependent on some trajectory prediction or location estimation techniques. The performance of these schemes highly depends on the accuracy of trajectory prediction or location estimation algorithms. A large number of active sensor nodes are also used. Comparing with these schemes, our approach does not try to predict the accurate trajectories, but to predict the possible moving area. It greatly relieves the dependence on the accuracy of trajectory prediction algorithm. The focus of this paper is to design proper sensor selection scheme based on moving area prediction to balance the energy consumption and tracking quality. The work on the kinetic model of moving target [23] can be helpful to design high accuracy trajectory predicting algorithms. And it can be integrated into our scheme to improve the performance.

RARE-Area and RARE-Node algorithms [13] for target tracking may reduce the number of nodes participating in tracking and so increase the energy efficiency. However, their approach assumed that a single node close to a target can detect the status of the target. It is a too strong assumption.

To get the insight into the energy consumption and tracking quality, a study on power-centric sensor deployment schemes that are independent of tracking methods and collaboration protocols is performed in [12]. The notion of quality of surveillance and tracking is introduced and used to guide the protocol design. The trade-off in surveillance phase and tracking phase is analyzed. They also proposed PECAS method for the tracking. Controlled Greedy Sleep (CGS) algorithm is a quasi-optimal synchronized sensor scheduling algorithm which increases network lifetime while maintaining correct functionality, based on local decisions of sensors [5]. These results are all based on coverage.

Coverage is an important criterion in devising target tracking application. It affects the quality of monitoring in the operational field. The centralized solutions based on approximation techniques [6,30] or on integer programming [8] are proposed to determine the minimum set of sensor for covering every location in the target field. Cardei and Wu [6] have reviewed different coverage models and solutions. Current research efforts on coverage [6,16,33] have focused on full coverage of the target field. Generally, high coverage means high tracking quality in the tracking applications. However, full high coverage limits the nodes selection. Some nodes will be used in tracking mostly, and their energy will be consumed quickly, which may lead to energy hole. In many cases, the deployment cost, the physical limitations, and the operational efficiency make the full high coverage fail.

Probabilistic coverage [2,3] gives us more options. Sheu and Lin [29] and Hefeeda and Ahmadi [14] show probabilistic coverage based solution can reduce the energy consumption and improve the performance of sensor network based applications. Huang and Tseng [16] lay a foundation for testing network p-coverage solely based on local information. Ren et al. [26] uses analytical model to demonstrate that probabilistic coverage can balance object-tracking quality and network lifetime. However, using the probabilistic coverage in the tracking applications directly may not be an efficient way. This is because there are more candidate nodes than actually needed for tracking operations. Proper sensor scheduling scheme can select the nodes optimally from the candidate set to balance energy consumption and tracking quality.

3. System overview and model

3.1. System overview

Our researches focus on moving target tracking application. We assume that the sensor nodes are scattered randomly in a geographical region. Each sensor is aware of its location. Location information can be gathered using an on-board GPS receiver. Absolute location information is, however, not needed. It is sufficient for the sensors to know their location with respect to a common reference point. Many localizing techniques can be used with varying degree of hardware complexity and accuracy. The sensor nodes are stationary in our model; this makes the localization problem somewhat simpler. Since the work presented here is not dependent on any particular localization method used, we do not emphasize any particular technique. The sensors must be capable of estimating the distance of the target to be tracked from the sensor readings. It is assumed that the sensor has already learned the sensor reading to distance mapping.

The system adopts a distributed architecture where the sink nodes are responsible for the initialization tracking system (such as time synchronization), visualization of the target trajectory and maintenance the database, whereas sensor nodes are responsible for local target detection, and target track data association.

The goal of this paper is to develop techniques for the above moving target tracking problem. The focus of our study is to design an energy-efficient sensor scheduling algorithm in order to balance energy consumption and tracking quality. For the sake of simplicity, we focus on the collaborative sensor scheduling algorithm during the tracking phase, ignoring the detection phase and glossing over the details of routing the query into regions of interest. We further assume there is one leader node active at any moment, and its task is to perform tacking algorithm and sensor scheduling algorithm, and route tracking information to the next leader. Leader nodes perform sensor selection algorithm to decide which sensors to be awake at next time point, and then transmit tracking information to sink nodes periodically. If the leader node does not have enough power to perform scheduling task., it transmit tracking information to sink node, and sink node selects a new leader node to perform this scheduling task or does it itself.

3.2. Problem definition

We consider the issue of tracking moving target with certain level of tracking quality, while conserving power. We address the target tracking problem by taking into account both coverage problem and minimizing energy consumption.

Coverage is an important issue in target tracking application. On one hand, the more intensive a moving target is covered by sensors, the more possible it is tracked accurately. On the other hand, though the moving target is in the sensing range of sensor i at certain time point, it may not be monitored by sensor because of the possible instance that sensor i dies out due to energy depletion or low data availability due to signal attenuation and noises. Thus, coverage should not be the only criterion in devising target tracking application.

Obviously, considering only coverage criterion is not enough to obtain certain level of tracking quality, communication cost and sensor's availability both needed to be considered. The sensor scheduling schemes are aiming to select a sensor subset which not only maximize the lifetime of the network but also provide certain tracking quality guarantee. The sensor scheduling scheme can be accomplished as follows:

(1) Predict the possible moving area of the target at next time point, X_{t+1}.
(2) Decide the candidate sensors covering the possible moving area of the target using α–k-coverage algorithm.
(3) Select a suitable sensor subset S' from the candidate sensor set S, in order to minimize the communication cost, thus to minimize the energy consumption.

The goal of this scheme is to schedule the activities of sensor nodes such that the target can be continuously observed under certain quality guarantee and network lifetime is maximized.

3.3. Energy consumption model

Sensors use energy to run circuitry and send radio signals. The later is usually a function of distance and takes a large potion of the energy. Radios typically have four power levels corresponding to the following states: transmitting, receiving, listening, and sleeping. Typically, the power required to listen is about the same as the power to transmit and receive. The sleep power is usually one to four orders of magnitude less. For Mica2 Mote sensors [1], these power levels are: 81 mW for transmit, 30 mW for receive and idle, 0.003 mW for sleep. Thus, a sensor should sleep as much as possible when it is not engaged in communication. We assume that the mean energy consumption rate of sensor node can be measured. In this paper, we propose the following energy consumption model.

The mean energy consumption rate of node i, denoted as w_i, is

$$w_i = e_p + e_l + e_t = e_p + \lambda_l l_i + (\lambda_t/|A_i|)\sum_{j\in A_i}^{A_i} d_{ij} \tag{1}$$

where e_p is the energy for data processing per unit time, l_i is the distance between node i and target location, e_l is the energy for sensing per unit time, e_t is the energy for data communication with neighbor nodes per unit time, A_i is the set of neighbor nodes of sensor node i, $|A_i|$ is the number of neighbor nodes of sensor node i, d_{ij} is the distance between sensor node i and sensor node j, λ_l (J/s m) is the energy for measuring per unit distance per unit time, λ_t (J/s m) is the energy for communication per unit distance per unit time. Parameters λ_t and λ_l mostly depended on the characteristic of sensor node. We assume that λ_t and λ_l are constants known. Assuming each node i has an initial battery energy E_i, the lifetime T_i of node i is defined as the expected time for the battery energy E_i to be exhausted, that is, $T_i = E_i/w_i$ where w_i is given by (1).

4. Probabilistic-based dynamic area coverage

In this section, an approximate prediction approach is developed to predict the possible moving area of target at next time point. Though we cannot predict the next location of target accurately, we can predict the possible moving area of target at next time point, that is, we can predict the approximation bound of the next location of target. Then, we propose a probabilistic-based dynamic non-complete k-coverage method, α–k-coverage, which can guarantee that target moving area is covered by at least k sensors under at least α probability. Thus, a candidate sensor set can be selected using α–k-coverage algorithm.

Proposition 1. *Assume that state estimation uncertainty of a moving target can be approximated by a Gaussian distribution. Then, the uncertainty of the target state estimation at next time can be illustrated by an uncertainty ellipsoid in the state space.*

Proof. Let T is time step and $X^t = [x^t \quad y^t]^T$ is the state of a target at time t, velocity vector is

$$V^t = [\, v_x^t \quad v_y^t \,]^T.$$

The state equation is $X^{t+1} = X^t + V^t T$.

$$\text{Let } V \sim N(\mu, \Sigma), \mu = (\,\mu_x \quad \mu_y\,)^T, \sum = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}, \text{ then, } E(X^{t+1}) = \begin{pmatrix} \hat{x}^t + \mu_x T \\ \hat{y}^t + \mu_y T \end{pmatrix}, var(X^{t+1}) = T^2 \sum,$$

where $(\,\hat{x}^t \quad \hat{y}^t\,)^T$ is the target's state estimated by Extended Kalman Filtering at time t, $E(X^{t+1})$ is the expectation of X^{t+1}, and $var(X^{t+1})$ is the covariance of X^{t+1}. Thus,

$$X^{t+1} \sim N(E(X^{t+1}), T^2\Sigma) = N\left(\begin{pmatrix} \hat{x}^t + \mu_x T \\ \hat{y}^t + \mu_y T \end{pmatrix}, T^2\Sigma\right).$$

Let $Z = f(X^{t+1}|X^t)$, then,

$$(X^{t+1} - EX^{t+1})(T^2\Sigma)^{-1}(X^{t+1} - EX^{t+1}) = -\ln(4\pi^2 Z^2 T^2 \Sigma)$$

$$\frac{\sigma_x^2\sigma_y^2}{T^2(\sigma_x^2\sigma_y^2 - \sigma_{xy}^2)}\left[\frac{(x^{t+1} - a)^2}{\sigma_x^2} - \frac{2\sigma_{xy}}{\sigma_x^2\sigma_y^2}(y^{t+1} - b)(x^{t+1} - a) + \frac{(y^{t+1} - b)^2}{\sigma_y^2}\right] = -\ln(4\pi^2 Z^2 T^2 \Sigma), \tag{2}$$

where Z is a constant which is in range $(0,1]$. For simplicity, we Let $\hat{x}^t + \mu_x T = a$, $\hat{y}^t + \mu_y T = b$.

The problem is divided into the following two cases.

Case 1. $\sigma_{xy} = \mathrm{cov}(x, y) = 0$, Eq. (2) is transformed to

$$-\ln(4\pi^2 Z^2 T^2 \Sigma) = \frac{1}{T^2}\left[\frac{(x^{t+1} - \hat{x}^t - \mu_x T)^2}{\sigma_x^2} + \frac{(y^{t+1} - \hat{y}^t - \mu_y T)^2}{\sigma_y^2}\right]$$

This case can be further divided into two sub-cases according to V.

Case 1.1. $V \sim N(0, I_2)$, then

$$(x^{t+1} - \hat{x}^t)^2 + (y^{t+1} - \hat{y}^t)^2 = -T^2 \ln(4\pi^2 Z^2 T^2 \Sigma)$$

X^{t+1}, the target's state uncertainty at next time can be illustrated by an uncertainty circle in the state space, and the center of the circle is the state estimation $\widehat{X}^t$ at time t, its radius is

$$T\sqrt{-\ln(4\pi^2 Z^2 T^2 \Sigma)}.$$

Fig. 1 shows the uncertainty ellipsoid in this sub-case.

Case 1.2. $v_x \sim N(0, \sigma_x^2)$, $v_y \sim N(0, \sigma_y^2)$, then

$$\frac{(x^{t+1} - \hat{x}^t)^2}{\sigma_x^2} + \frac{(y^{t+1} - \hat{y}^t)^2}{\sigma_y^2} = -T^2 \ln(4\pi^2 Z^2 T^2 \Sigma)$$

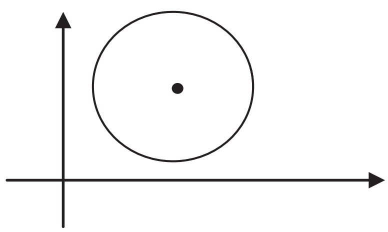

Fig. 1. The results of case 1.1.

X^{t+1} can be illustrated by an uncertainty ellipsoid in the state space, and the center of this ellipsoid is the state estimation $\widehat{X}^t$ at time t. The long half axis is

$$\sigma_x T\sqrt{-\ln(4\pi^2 Z^2 T^2 \Sigma)}$$

and short half axis is

$$\sigma_y T\sqrt{-\ln(4\pi^2 Z^2 T^2 \Sigma)}$$

(if $\sigma_x > \sigma_y$). Fig. 2 shows the uncertainty ellipsoid in this sub-case.

Case2: $\sigma_{xy} = \mathrm{cov}(x, y) \neq 0$

X^{t+1} can be illustrated by an uncertainty ellipsoid in the state space, and the center of this ellipsoid is $\left(\hat{x}^t + \mu_x T \quad \hat{y}^t + \mu_y T\right)^T$. Fig. 3 shows the uncertainty ellipsoid.

From all above cases, it can be saw that, the uncertainty of the target state estimation at next time can be illustrated by an uncertainty ellipsoid in the state space. The area of this uncertainty ellipsoid is depended on the time step T, the covariance of velocity and density function constant Z. □

In fact, the area of the uncertainty ellipsoid at any time point t indicates the error of state estimation or state prediction. Thus, the ellipsoid area at time point t can be approximately as the possible moving area of target at time point t. All sensors which cover the ellipsoid area become a candidate sensor set using certain coverage algorithm.

In traffic engineering, it has widely been accepted that the speeds in the free-flow vehicle traffic state can be considered as normally distributed [27]. And under free-flow conditions, the walking speeds of pedestrians in the certain spots like airports and subway terminals can also be considered as normally distributed [39]. This is the reason why the Gaussian distribution is adopted here to predict the next moving area.

For tracking a moving target with non-Gaussian distribution, for example, wild animals tracking, we propose the other approach to predict target moving area. First, we predict the next location of target, X^{t+1}, using a certain predicting method. Then, as Fig. 4 shows, the target possible moving area at next time can be illustrated by an uncertainty circle in the state space, and the center of the circle is the predicted next location X^{t+1}, its radius is $r' = r/2$, where r is the sensing radius of sensor node.

For a target tracking sensor network, partial coverage may be enough to provide certain tracking quality with less energy consumption compared with full coverage. A relaxed sensing coverage may be more appropriate for balancing target-tracking quality and battery power consumption. A probabilistic-based dynamic non-complete k-coverage method, α–k-coverage, which guarantees that target moving area is covered by at least k sensors under at least α probability, is adopted in our scheme.

We assume that the possible area of target's state prediction at next time is X_{t+1}, the area of X_{t+1} is $S(X_{t+1}) = S_{t+1}$. The partial area of X_{t+1} which is covered by sensor s_i is A_i, that is, $S(X_{t+1} \cap s_i) = A_i$. Then, the probability p_i that the moving target is covered by sensor s_i at next time is $p_i = S(X_{t+1} \cap s_i)/S(X_{t+1}) = A_i/S_{t+1}$. As Fig. 5 shows, the intersection areas of sensor s_1, s_2, s_3 and the target's moving area at next time are A_1, A_2, A_3. Then the probabilities that the moving target is covered by the three sensors are $p_1 = A_1/S_{t+1}$, $p_2 = A_2/S_{t+}$, $p_3 = A_3/S_{t+1}$. Obviously, $p_1 < p_2 < p_3 = A_3/S_{t+1} = 1$.

Definition 1. Given a probability α and a target area X_{t+1}, the probability that the moving target is covered by sensor s_i at next time is p_i, then, the candidate sensor set at next time point is $S = \{s_i | p_i \geqslant \alpha, i = 1, 2, \ldots, k\}$.

(i) In the case $\alpha = 1$, the area X_{t+1} of target's state prediction is entirely covered by k sensors of set S, that is, the target area X_{t+1} is k-covered.

(ii) In the case $0 < \alpha < 1$, the area X_{t+1} of target's state prediction is covered by k sensors of set S at least probability α, conditionally on this, we define the target area X_{t+1} satisfying no-complete k-coverage, α–k-coverage.

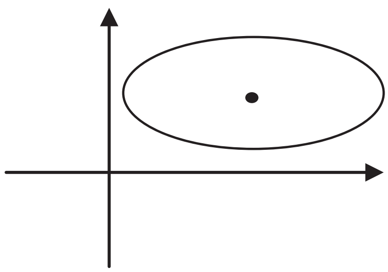

Fig. 2. The results of case 1.2.

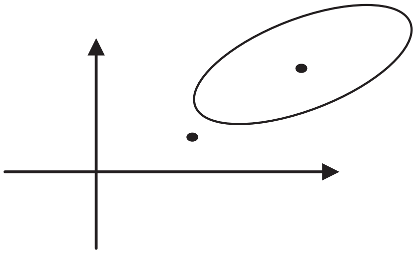

Fig. 3. The results of case 2.

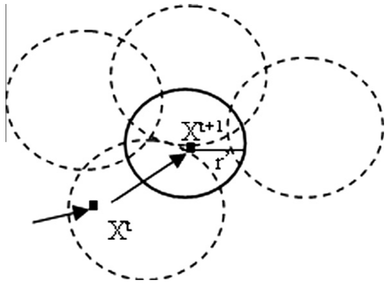

Fig. 4. The possible moving area at next time point.

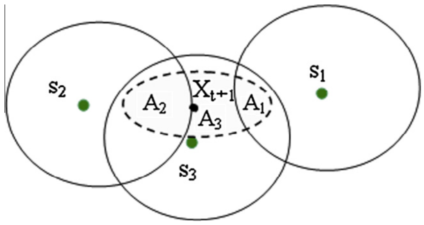

Fig. 5. An example of α–k-coverage.

As the probability α is small, more sensors can meet α–k-coverage requirement, that is, k will be a greater value. When α increases to a greater value even close to 1, α–k-coverage is approximately k-coverage, fewer sensors can meet requirement, so k will be a small value. In fact, given the certain predicted area, α and k are approximately inversely proportional.

5. Energy-efficient sensor scheduling algorithm

A novel sensor scheduling algorithm, Optimal Cooperation Scheduling Algorithm (OCSA), is proposed in this section to select a suitable sensor subset from the candidate set by both taking into account energy cost and sensor's availability. The goal is to find a suitable subset of sensors which not only minimize the transmission energy but also provide certain tracking quality guarantee. The whole tracking process is divided into a series of time step. In each step, the candidate set is determined by probabilistic coverage and the tracking sensors are selected by OCSA

We formulate the Sensor scheduling problem as follows:

Given:

- A candidate sensor set $S = \{s_i | p_i \geqslant \alpha, i = 1, 2, \ldots, k\}$ in t-th time step.
- The mean energy consumption rate of sensor i, w_i, for all $s_i \in S$.
- The residual energy of sensor s_i in t-th time step, $E_{i,t}$, for all $s_i \in S$.
- The least energy limitation E for consumption within each time interval.

Find:

- The selected optimal sensor subset in t-th time step, S'.

Variables:

- x_i, Boolean variable, for $i = 1,2,\ldots,k$; $x_i = 1$ if sensor $s_i \in S'$, otherwise $x_i = 0$.

The total energy cost including sensing and communicating cost of sensor node i in t-th time step is defined as

$$C_{i,t} = \beta x_i l_i + (1-\beta)x_i g(x_i) \tag{3}$$

where $g(x_i) = \Sigma_{j\neq i}^{k}(x_j d_{ij})/\Sigma_{j\neq i}^{k} x_j$.

Since we do not know the exact target's location, we use the center of predicted ellipsoid/circle as the expected target location. l_i is the distance between sensor i and expected target location. $\beta x_i l_i$ is the energy cost for sensor i to measure the target and determine the relative distance. β is energy consumption ratio for measuring, $(1-\beta)x_i g(x_i)$ is the mean energy cost for communication with neighbor sensors under the condition that sensor i is selected. In fact, β depends on sensor sensor's characteristics, such as transmission range and sensing range. Then, the total energy cost of sensor nodes in the candidate sensor set S in t-th time step is,

$$f(x_1, x_2, \ldots, x_k) = \Sigma_i^k C_{i,t} \tag{4}$$

The optimization problem can be written as:

$$\begin{aligned}
\text{Minimizing } & \sum_{i=1}^{k} C_{i,t} \\
\text{subject to } & (E_{i,t} - w_i \Delta t)x_i \geqslant 0 \text{ for } i = 1,2,\ldots,k \\
& \Sigma_i^k E_{i,t} x_i \geqslant E \\
& \Sigma_i^k x_i \geqslant N_L
\end{aligned}$$

where $x_i = 0$ or $x_i = 1$.

Remarks:

- The first constraint, $(E_{i,t} - w_i \triangle t)x_i \geqslant 0$ for $i = 1,2,\ldots,k$, guarantees that the residual energy $E_{i,t}$ of sensor s_i is enough for tracking within next time step. $\triangle t$ is the length of time step.
- The second constraint, $\Sigma_i^k E_{i,t} x_{i,} \geqslant E$ guarantees that total residual of selected sensors is not less than the energy limitation E. According the energy consumption model, combining different sensors to cooperate tracking may lead to different energy consumption. A total energy limitation can help to combine suitable sensors.
- The third constraint, $\Sigma_i^k x_i \geqslant N_L$ guarantees that at least N_L sensors will be selected. N_L is the number of sensors necessary to locate the target.

The goal is to find the sensors from the candidate sensor set that are close to the target and close to each other to minimize the measuring and communicating cost. Both $\beta x_i l_i$ and $(1-\beta)x_i g(x_i)$ depend only on whether the nodes are selected or not in the t-th time step. Therefore, the optimizing problem is a binary linear problem, which is classified as NP-hard. Most existing methods are too complex to be implemented on resource limited sensors. In this section, we propose a Greedy approach to release the computing burden needed to solve this optimizing problem. Our heuristic algorithm takes the candidate sensor set S that contains k sensors determined by α–k-coverage in time step t as the input parameters, and returns a suitable sensor subset S', $S' \subseteq S$.

Let the residual energy of k sensors in set S are $E_{1,t}, E_{2,t}, \ldots, E_{k,t}$ in the beginning of t-th time step and sort the sensors according to the residual energy in ascending order, which means that $E_{1,t} \leqslant E_{2,t} \leqslant \ldots \leqslant E_{k,t}$. We define $X_{pre} = (x_1, x_2, \ldots, x_k)$ is the last candidate scheduling strategy, and define $X_{new} = (X_{pre}; x_j = 1)$ is a new candidate scheduling strategy. X_{pre} is initially set the as ($x_i = 0$, $i = 1,2,\ldots,N_L$, $x_i = 1$, $i = k - N_L + 1,\ldots,k$). Since the sensors are sorted according to the residual energy in ascending order, the initial schedule strategy is to select N_L sensors with the most residual energy. The operation (X_{pre}; $x_j = 1$) replaces the selected sensor that has the most residual energy with sensor j.

Wireless sensor networks are characterized by limited onboard energy supply. It is important to make the sensors' energy consumption balance; otherwise the sensors carrying more communicating or sensing tasks will deplete their energy budget faster than other sensors and die quickly, which drastically reduce the useful lifespan of sensor networks. This uneven energy depletion phenomenon is called "energy hole". In target tracking sensor networks, if the sensors that are close to the target moving area and close to each other are always selected, their energy consumes faster and they will not be available for tracking and communicating quickly. In the rest of tracking period, the solution space is narrowed down and the solution quality decreases. Besides minimizing the energy consumption, making the energy consumption even is also very important. The idea of our paper is using probabilistic coverage to get more candidate sensors and avoid "energy hole" as possible as it can. So OCSA-Greedy selects the sensors with the most residual energy first in each time step, and then tries to reduce the energy cost by considering the other sensors with less residual energy. If it can lead to a lower energy cost, we update the selection set.

Algorithm 1: **(Greedy-OCSA Heuristic)**

Input: A candidate sensor node set S, time step t
Output: A sensor subset S'.
start = $k - N_L$
While start >1 **do**
for all the forward start-1 sensor nodes **do**
  if the j-th sensor 's residual energy is enough **then**
  Computer objective function $f(X_{pre};\ x_j = 1)$ suppose $x_j = 1$
  end if
end for
find the least $f(X_{pre};\ x_j = 1)$ from $f(X_{pre};\ x_j = 1), \ldots, f(X_{pre};\ x_{start-1} = 1)$
Computer total residual energy of the candidate strategy $(X_{pre};\ x_j = 1)$, Energy_total
if $f(X_{pre};\ x_j = 1) > f(X_{pre})$ && Energy_total > E **then**
  X_{pre} is a suitable solution and stop search
 else then
  Select sensor j, $x_j = 1$.
 update the search place of the next iteration, start = j
end if
end while
return the corresponding sensor subset S' of strategy X_{pre}.

The complexity of the Greedy-OCSA Heuristic is $O((k - N_L)(k - N_L))$, where k is the number of candidate sensors and N_L is the number of sensors necessary to locate the target. The heuristic runtime is $O((k - N_L)(k - N_L))$.

An example that uses α–k-coverage and Greedy-OCSA to select the tracking sensor sensors is shown in Fig. 6. First, the next moving area is predicted as an ellipsoid. Then a candidate sensor set is determined by α–k-coverage, namely sensors 1–8 in this example. If we use k-coverage, the candidate set is narrowed to sensor 3, sensor 6, sensor 7 and sensor 8. Only these sensors will be used in tracking, and their energy will be consumed quickly, which may lead to energy hole. If we used α–k-coverage, the larger size of candidate set provides more opportunities for optimization. Not all the sensors are necessary for the tracking. Greedy-OCSA selects the sensors with more residual energy and less measuring/communicating costs to track the target. Here they are sensor 3, sensor 6 and sensor 7. Therefore, more sensors are involved in the tracking process and their energy consumption can be balanced. The whole network lifetime may be prolonged.

6. Performance evaluation

Performance evaluation is done through intensive simulation. NS-2 is the de facto general network simulator. However, NS-2 does not work well for large topologies (more than 300 nodes) and the built-in routing algorithms could be buggy [15,28]. Since our goal is not to design a protocol, but to do a detailed study of tradeoff analysis between power consumption and tracking quality, a simulator is developed specifically for our study to simulate the moving target tracking environment and evaluate the proposed sensor scheduling scheme.

6.1. Simulation setup

Table 1 summarizes the system parameters and their settings. The network field is a square of size of 500 m × 500 m. We use 100 sensor nodes with uniformly random distribution. The target appears in the field at a random location. We generate target's velocity with a Gaussian distribution. The average value is 20 m/s and the variance is 10 m/s. We adopt the sensor's

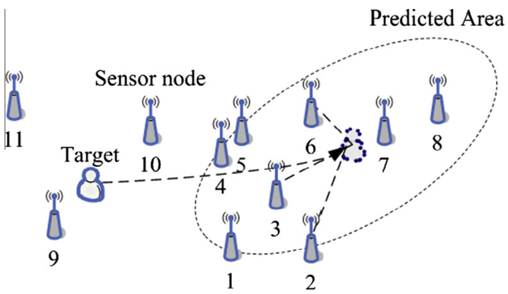

Fig. 6. An example of Greedy-OCSA.

Table 1
System parameters and settings.

Parameter	Setting
Number of sensor nodes	100
Initial energy at each sensor	0.1 KJ
Power consumption in sleeping mode	0.01 J
Energy consumption rate	$\lambda_l = 0.1$, $\lambda_t = 0.5$
Epoch	20 time unit
Sensor sensing radius	80 m
Sensor transmission radius	80 m
Sensor network field	500 m $\times$ 500 m

energy model described in the Section 3.3. The parameters can be estimated from battery characteristics. The initial energy of a sensor is set to 0.1 KJ, which means this sensor has two 1.5 v batteries with 100 mAh capacity. Energy consumption rate is set according to existing measurement.

In the simulation we consider the following tunable parameters:

- α, the least probability that the target is covered at next time. We use α–k-coverage method to decide the candidate sensor set. We change α from 0.1 to 0.6 to study the effect of network coverage on target tracking quality. The initial value is set to 0.2.
- β, the energy consumption weight for measuring the distance between sensor and target. Parameter β-depends on sensor's characteristics. Generally, sensor's transmission range is more than its sensing range. For different sensors, such as infrared sensors and ultrasound sensors, β has different range because of different sensing ranges. Generally, passive infrared sensor has less sensing range than other sensor, so β should increase to improve tracking accuracy. We change β from 0.1 to 0.75 to study the effect of sensing range for different sensor's characteristics on target tracking quality. The initial value is set to 0.4.

Both Extended Kalman Filtering (EKF) ($N_L = 2$) and maximum likelihood estimation (MLE) method ($N_L = 3$) are utilized to track moving target. We design an EKF using a state vector with four components, two position components (x, y) and two velocity components (v_x, v_y). We found that in the initial phase the performance of EKF is bad, that's because the performance of EKF highly depends on history information, so we use MLE is to initialize and reset the Kalman filter. The result shows that the combination of EKF and MLE is an appropriate approach. Fig. 7 shows the tracking result of a moving target using EKF and MLE. It can be seen that the tracking quality can be accepted.

To find the target, at the very beginning of tracking task, we must wake up all sensor nodes. Since we focus on balancing the energy consumption and tracking quality, we only record the data after the target is found. At this time, EKF is used to track the target, so N_L is set to 2. In the simulation, we generate 20 trajectories of moving target. For each trajectory, we simulate the tracking process 20 times.

6.2. Simulation results of probabilistic-based dynamic area coverage

First, we predict the possible moving area of target at next time point using the approximate approach proposed in Section 4. Fig. 8 shows the prediction result of uncertainty ellipsoid in the state space. Our simulation experiment

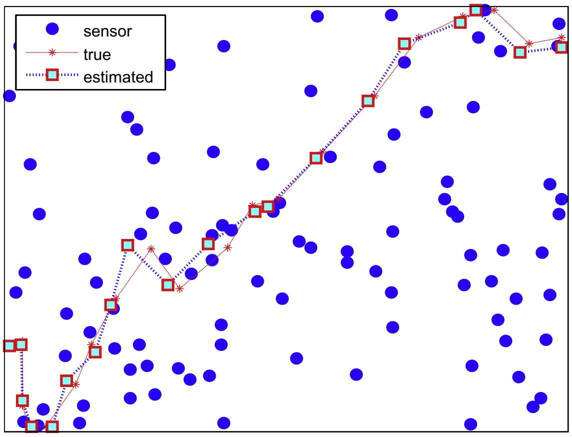

Fig. 7. A tracking result using EKF and MLE.

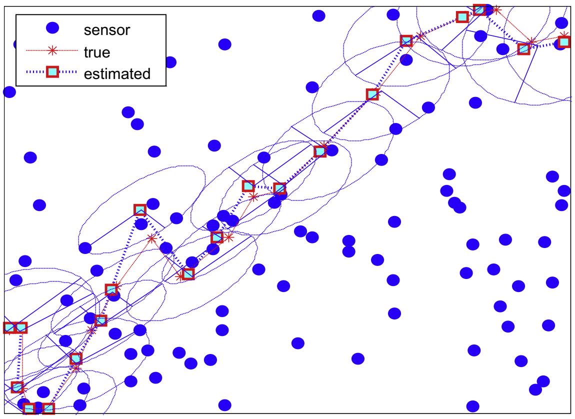

Fig. 8. The uncertainty ellipsoids of the next state.

demonstrates that prediction of the next location is not always accurate. However, the location of target at next time point always falls into our predicted moving area. Based on our simulation, the ratio that the target gets into the predicted moving area is 100%.

Then, a candidate sensor set S can be decided using α–k-coverage method. In our simulation, k is set to 6. Fig. 9 shows an example of 0.3–6-coverage, where $\alpha = 0.3$, $k = 6$. The solid circles with different colors represent the sensing area of sensors that have 0.3-coverage on the target moving area (the ellipsoid drawn with dotted line).

It is an interest problem that how much history information is needed to predict the possible moving area of target at next time point when the scheduling algorithm is performed in general sensor node. When a leader sensor has no enough information to predict the next moving area, it can send request message to sink node and get enough information from sink node, it can also broadcast request message to its neighbors and get former tracking information within h hops range. The hop number h can be decided through communication distance, measurement distance and the acreage of target region. In our simulation, we adopt the latter method and set h to 3. It is because getting the information from neighbor nodes may be more time-efficient than getting information from sink node. Additionally, sink node may become bottleneck if we use the former method. How to set h value to get more information with less overhead may be an interesting problem needing further research.

6.3. Simulation results of sensor scheduling

In order to evaluate the performance of Greedy-OCSA and compare Greedy-OCSA to existing schemes, we have both implemented Greedy-OCSA approach and CGS [5]. The motivation of CGS is similar to ours. This is the reason why we choose CGS. Instead of developing a totally new effective tracking algorithm, we focus on achieving better tradeoff between power consumption and quality of surveillance through moving area prediction and coverage-based sensor selection. In order to demonstrate the advantage of combining α–k-coverage and Greedy-OCSA, we also compare our approach with k-coverage and k-coverage with Greedy-OCSA.

As shown in Fig. 10, k-coverage approach causes the largest number of sensors in tracking mode since k sensors are selected in every step. CGS and k-coverage with Greedy-OCSA can reduce the number of tracking sensors by adopting certain scheduling schemes. Our proposed approach uses the smallest number of sensors for tracking. The average value is 3.75,

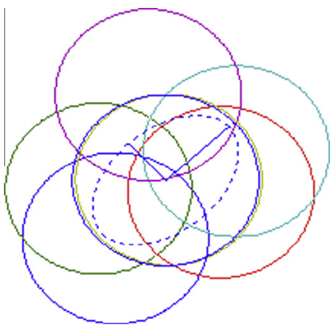

Fig. 9. α–k-Coverage.

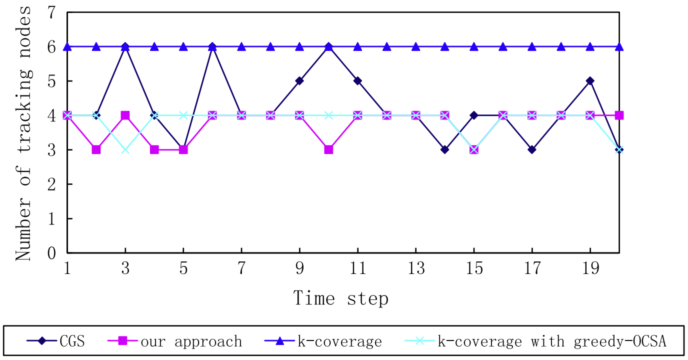

Fig. 10. The number of sensors in tracking mode.

a 15% improvement compared with CGS and a 3% improvement compared with *k*-coverage with Greedy-OCSA. It is because our approach has a biggest candidate set for optimizing sensor selection thankful to α–k-coverage.

Generally, less sensors are active, more energy is saved. However, the number of active sensors cannot tell the detail energy dissipation. For example, if a sensor is always active, it exhausts its energy soon. This will cause energy hole and make the whole sensor network fails. Besides number of active sensors, we also estimate the average and lowest value of residual energy ratio to describe the energy consumption.

As shown in Fig. 11, the average residual energy ratio reflects the trend that less tracking sensors cause less energy consumption. This is the reason why *k*-coverage leads to the highest energy-consumption. Our approach is still the best one. Although the difference is not very significant, our approach get 5% gain compared with *k*-coverage with Greedy-OCSA, 6% gain compared with CGS and 8% gain compared with *k*-coverage.

The lowest residual energy ratio given in Fig. 12 tell us serious unbalanced energy consumption will happen without proper scheduling schemes. With *k*-coverage, a sensor consumes 60% energy in the worst case. It means some sensors may exhaust their energy quickly and leads to energy hole, which ultimately causes the whole sensor network fail. By using Greedy-OCSA, *k*-coverage with Greedy-OCSA can reduce the largest energy consumption by 8%. CGS uses the probe-and-sleep mechanism to make the worst consumption reduce 25%. The worst consumption ratio is only 0.35 in our approach. It is because our approach selects the tracking sensors from the biggest candidate set.

Fig. 13 shows the tracking quality of these four approaches. The tracking quality is measured by error that indicates the distances between true location and estimated location at each time point. It is given in the form of Mean Square Error (MSE). It can be seen that *k*-coverage has the best performance in this metric. The reason is that *k*-coverage wakes up the largest number of sensors to track target in these four approaches. It comes with the highest and unbalanced energy consumption. The difference between *k*-coverage and our approach is not significant. Our approach does not miss the target at any time point. The error is still in a reasonable level. And our approach is better than CGS and *k*-coverage Greedy-OCSA. The CGS cause the highest error since its probing and sleeping does not consider the measuring and communicating cost. Instead of randomly making some sensors sleeping, Greedy-OCSA implements close-form optimization of sensor selection.

The simulation results show that, with the same parameter settings, our approach performs better than CGS, *k*-coverage and *k*-coverage with Greedy-OCSA. Our approach can get better trade-offs between energy consumption and tracking quality.

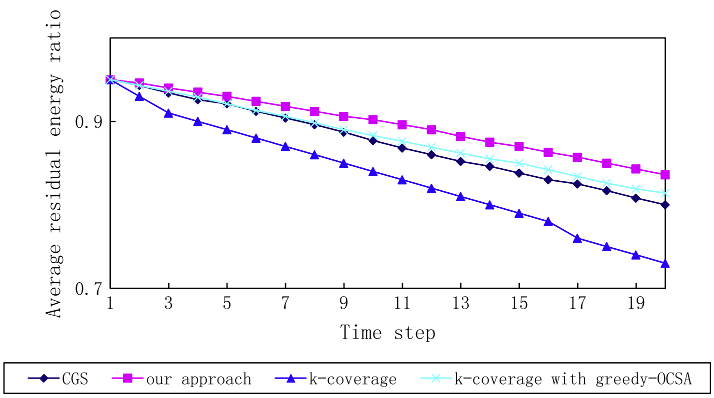

Fig. 11. The average residual energy ratio.

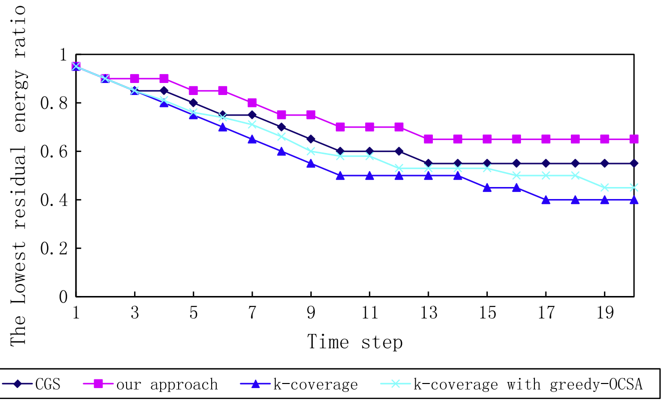

Fig. 12. The lowest residual energy ratio.

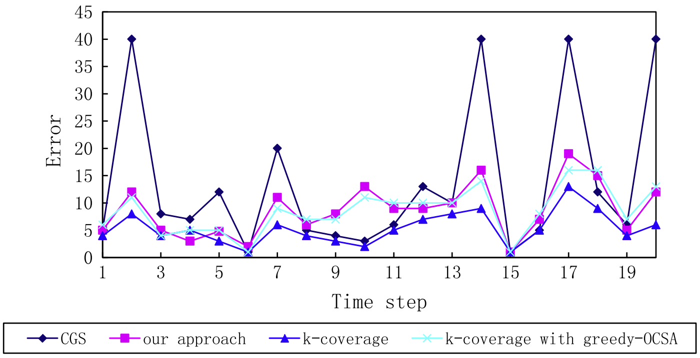

Fig. 13. The tracking error.

Since Greedy-OCSA is a close-form solution to binary linear optimal problem that may not find a global optimum, we also use branch and cut approach [17] to solve this problem and compare their results. Since they both use probabilistic coverage and share the same constraints, they get the same tracking qualities. Greedy-OCSA consumes more energy than branch and cut as shown in Fig. 14, in which the average consumed energy ratio is used as the metric instead of the average residual ratio used in the previous figure to show the performance difference more clearly. For example, in the first step, Greedy-OCSA consumes 5% of the total energy, and branch and cut consumes 4.15% of the total energy. If residual energy ratio is used, the result is 95% vs. 95.85%. This difference is too small to be shown clearly. The average difference between consumed energy is 16%. In the early stage, when the nodes have more power, the Greedy-OCSA is 17% worse than branch and cut in energy consumption. After that, when the nodes consumes a lot of power, the difference becomes less significant. The

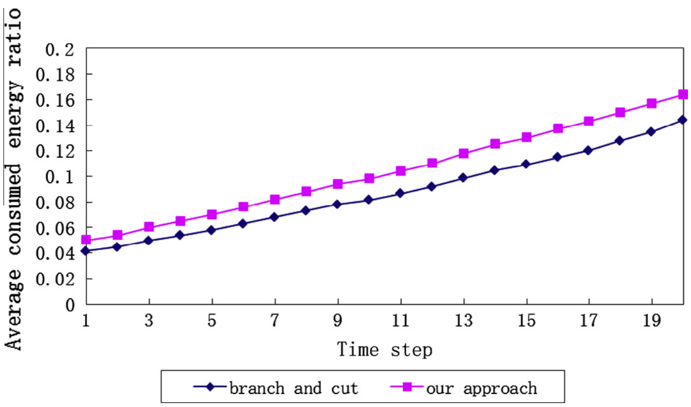

Fig. 14. Comparison with branch and cut.

consumed energy ratio of Greedy-OCSA is 16.4%, only 12% worse than the ratio of branch, 14.4%, in the last time stamp. It proves OCSA-Greedy can reduce the energy consumption in an even manner to ease the impact of "energy hole" on the performance. We will continue to improve the Greedy-OCSA in the future work.

6.4. Impacts of α and β

We also evaluate the performance of our approach by changing the parameter, α and β. Fig. 15 shows the impacts on the tracking quality. It can be found that as α increases to certain value (in this example it is 0.2), the MSE tends to reduce. However, as α increases to greater value, the MSE tends to increase and the quality of tracking decreases. The reason for this is that, as α increases, the coverage is more intensive, so that the candidate sensor set decided by α–k-coverage is closer to k-coverage, and few sensors can be selected as candidates, which may lead to "energy hole". However, if α is too low, many sensors including those that only have a little sensing coverage are also selected as candidates. This will degrade the performance of Greedy-OCSA.

Fig. 15 also shows the effect of the energy consumption weight β on target tracking quality. In fact, parameter β reflects the relation between sensor's transmission range and its sensing range for different sensor's characteristics. As shown in Fig. 15, as β increases to greater value, the MSE increases. The reason for this is that when weight β increases, the sensing range increases, similarly the sensor density increases, the more energy is consumed for measuring, so that some sensors exhaust energy quickly, so the tracking quality descends. It is also obvious that when weight β decreases to small, the MSE increases. The reason is that the sensing range decreases, there is no enough information to track target.

Fig. 16 shows the total energy consumption as parameters α and β are changed. As shown in Fig. 16, as α is small, the total energy consumption of all sensors is at a relatively high level, as α increases to a greater value, the total energy consumption reduces, but the MSE increases. The reason is that when α is small, more sensors are selected and awaked according to α–k-coverage requirement. When α increases to a greater value even close to 1, α–k-coverage is approximately k-coverage, fewer

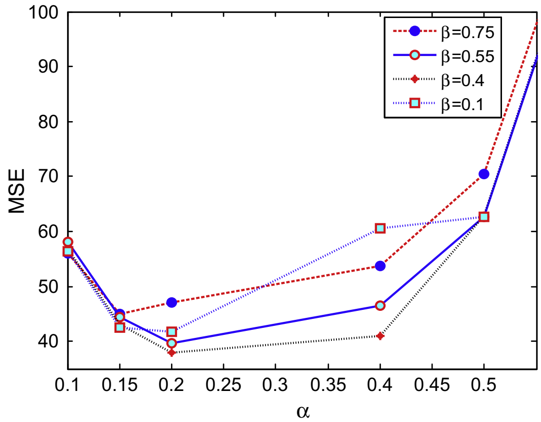

Fig. 15. The MSE vs. α and β.

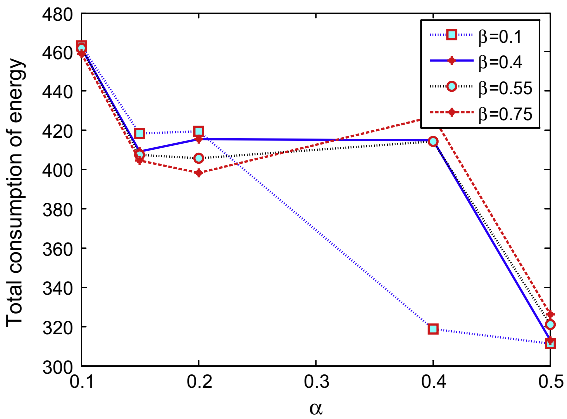

Fig. 16. The total consumption of energy vs. α and β.

sensors can meet requirement and be selected. Fig. 16 also shows that when α is small, variety of β has little effect on energy consumption because that more sensors can be selected and most energy is consumed by communication. When α increases to greater value, variety of β has big effect on energy consumption because that fewer sensors can be selected and the energy consumption of measuring increases. It can be found that when $\alpha = 0.2$, $\beta = 0.4$, there is good balance with MSE and energy consumption.

To summaries, α–k-coverage means the moving area is covered by at least k sensors under at least α probability. With high α value, the candidate set only contains few sensors close enough to the moving area. These sensors are always selected to track. The energy consumption is not even. The tracking quality will drop quickly after these sensors exhaust their power. If α is low, the candidate set becomes larger, more nodes can be selected to track and the energy consumption can be balanced. However, some nodes that only cover a small part of moving area may be selected, which could compromise the tracking quality and increase the energy cost. β reflects the relation between sensor's transmission range and its sensing range for different sensor's characteristics. Larger β value suggests larger sensing area. If β is too low, the sensing area is too small. The chance that detects and locates the target would be small too. If β is too high, the sensing area is bigger, which is good for detecting and locating targets. However, the energy consumption will increase significantly and reduce the lifetime of sensor networks.

The simulation result shows the proposed algorithm can balance the tracking quality and energy consumption with proper α and β value. We can get the small MSE with a reasonable energy consumption level. Thus, how to determine proper α and β value based on tracking environments is very important, we will address this issue in the future study. To demonstrate the feasibility of the proposed sensor scheduling algorithm, we are developing a prototype implementation for IMote2 Mote. Although this implementation is still in developing stage, the initial results are very positive. The average executing time (the time for leader election and handover is not including) is only 1.1 s, and the maximum executing time does not exceed 2 s. The average distance that targets move in this duration is only 30 m. We will continue this developing in future work.

7. Conclusion and future work

In this paper, we propose a probabilistic coverage based sensor scheduling scheme for target tracking sensor network. In this scheme, the possible moving area of target is predicted, and then α–k-coverage method is developed to guarantee that area is covered by at least k sensors under at least α probability. Those k sensors construct a candidate sensor set for tracking. The sensor scheduling problem is defined as finding a suitable sensor subset from the candidate set to balance the energy consumption and tracking quality. This problem can be formalized as a binary linear problem, which is classified as NP-hard. A heuristic algorithm, Greedy-OCSA, is proposed to solve this problem, which selects the sensors with the most residual energy first in each time step, and then tries to reduce the energy cost by considering the other sensors with less residual energy. Extensive simulations demonstrate the proposed sensor scheduling schemes can balance tracking quality and network lifetime under α–k-coverage condition. A prototype implementation is in developing.

In our future work, we will continue to develop our prototype implementation and evaluate our approach in practical environment. The issues such as optimizing the values of α and β, improving the performance of Greedy-OCSA, and determining the length of time step will also be addressed.

The source code of simulator is available for download:
http://keshi.iothust.org/simulation.html or
http://keshi.ubiwna.org/simulation.html.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant No. 51435009 and the Specialized Research Fund for the Doctoral Program of Higher Education under Grant No. 20110142110062.

References

[1] MICA2 Mote Datasheet <http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0042-05_A_MICA2.pdf>.
[2] N. Ahmed, S.S. Kanhere, J. Sanjay, Probabilistic coverage in wireless sensor networks, in: Proceedings of the IEEE Conference on Local Computer Networks, Sydney, Australia, November 15–17, 2005.
[3] H.M. Ammari, Stochastic k-coverage in wireless sensor networks, Wireless Algorithms Syst. Appl. 5682 (2009) 125–134.
[4] J. Aslam, Z. Butler, V. Crespi, G. Cybenko, D. Rus, Tracking a moving object with a binary sensor network, in: Proceedings of ACM International Conference on Embedded Networked Sensor Systems, Los Angeles, CA, USA, November 05–07, 2003.
[5] G. Bergmann, M. Molnár, L. Gönczy, B. Cousin, Optimal period length for the CGS sensor network scheduling algorithm, in: Proceedings of the Sixth International Conference on Networking and Services. IEEE, Cancun, Mexico, March 7–13, 2010.
[6] M. Cardei, J. Wu, Handbook of Sensor Networks, chapter Coverage in Wireless Sensor networks, CRC Press, 2004.
[7] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, J. Zhao, Habitat monitoring: application driver for wireless communications technology, ACM SIGCOMM Comput. Commun. Rev. 31(2 supplement), (2001) 20–41.
[8] K. Chakrabarty, S.S. Iyengar, H. Qi, E.C. Cho, Grid coverage for surveillance and target location in distributed sensor networks, IEEE Trans. Comput. 51 (12) (2002) 1448–1453.
[9] W.-P. Chen, J.C. Hou, L. Sha, Dynamic clustering for acoustic target tracking in wireless sensor networks, IEEE Trans. Mobile Comput. 3 (3) (2004) 258–271. special issue on Mission-oriented sensor networks.

[10] M. Chu, H. Haussecker, F. Zhao, Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks, Int. J. High Perform. Comput. Appl. 16 (3) (2002) 293–313.
[11] O. Demigha, W.K. Hidouci, T. Ahmed, On energy efficiency in collaborative target tracking in wireless sensor network: a review, IEEE Commun. Surv. Tutor. 15 (3) (2013) 1210–1222.
[12] C. Gui, M. Prasant, Power conservation and quality of surveillance in target tracking sensor networks, in: Proceedings of the 10th International Conference on Mobile Computing and Networking, Philadelphia, PA, USA, September 26–October 1, 2004.
[13] M. Guo, E. Olule, G. Wang, S. Guo, Designing energy efficient target tracking protocol with quality monitoring in wireless sensor networks, J. Supercomput. 51 (29) (2009) 131–148.
[14] M. Hefeeda, H. Ahmadi, Energy-efficient protocol for deterministic and probabilistic coverage in sensor networks, IEEE Trans. Parallel Distrib. Syst. 21 (5) (2010) 579–593.
[15] L. Hogie, P. Bouvry, An overview of MANETs simulation, Electronic Notes Theor. Comput. Sci. 150 (2006) 81–101.
[16] C.F. Huang, Y.C. Tseng, The coverage problem in a wireless sensor network, in: Proceedings of ACM Workshop on Wireless Sensor Networks and Applications (WSNA), San Diego, CA, USA, September 19, 2003.
[17] James J. Cochran, Louis Anthony Cox Jr., Pinar Keskinocak, Jeffrey P. Kharoufeh, J. Cole Smith, Wiley Encyclopedia of Operations Research and Management Science, Wiley, 2011.
[18] J. Li, Q. Jia, X. Guan, X. Chen, Tracking a moving object via a sensor network with a partial information broadcasting scheme, Inf. Sci. 181 (20) (2011) 4733–4753.
[19] Y. Liang, X. Feng, F. Yang, L. Jiao, Q. Pan, The distributed infectious disease model and its application to collaborative sensor wakeup of wireless sensor networks, Inf. Sci. 223 (20) (2013) 192–204.
[20] A. Mainwaring, R. Szewczyk, D. Culler, J. Anderson, Wireless sensor networks for habitat monitoring, in: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, Georgia, USA, September 28, 2002.
[21] A. Marculescu, S. Nikoletseas, O. Powell, J. Rolim, Efficient tracking of moving targets by passively handling traces in sensor networks, in: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 2008), New Orleans, LA, USA, November 30–December 1, 2008.
[22] K. Mechitov, S. Sundresh, Y. Kwon, G. Agha, Cooperative Tracking with Binary-Detection Sensor Networks, Technical Report UIUCDCS-R-2003-2379, Computer Science Dept., University of Illinois at Urbana-Champaign, 2003.
[23] B.A. Mohammad, A. Mohammad, A. Reza, K.S. Reza, A. Mahdlo, Modeling and simulation of trajectory in RADAR systems, in: Proceedings of the 4th Asia International Conference on Mathematical/Analytical Modeling and Computer Simulation, Kota Kinabalu, Borneo, USA, May 26–28, 2010.
[24] S. Pattem, S. Poduri, B. Krishnamachari, Energy-quality tradeoffs for target tracking in wireless sensor networks, in: Proceedings of 2nd Workshop on Information Processing in Sensor Networks (IPSN'03), Palo Alto, California, USA, April 22–23, 2003.
[25] H. Qi, Y. Xu, Mobile agent migration modeling and design for target tracking in wireless sensor networks, Ad Hoc Netw. 6 (1) (2008) 1–16.
[26] S. Ren, Q. Li, H. Wang, X. Chen, X. Zhang, A study on object tracking quality under probabilistic coverage in sensor networks, ACM SIGMOBILE Mobile Comput. Commun. Rev. 9 (1) (2005) 73–76.
[27] R.P. Roess, E.S. Prassas, W.R. McShane, Traffic Engineering, third ed., Prentice-Hall, Englewood Cliffs, NJ, 2004.
[28] B. Schilling, Qualitative Comparison of Network Simulation Tools, Modeling and Simulation of Computer Systems Seminar, Institute of Parallel and Distributed Systems (IPVS), University of Stuttgart, 2005.
[29] J.P. Sheu, H.Fu. Lin, Probabilistic coverage preserving protocol with energy efficiency in wireless sensor networks, in: Proceedings of the Wireless Communications and Networking Conference, Hong Kong, China, March 11–15, 2007.
[30] S. Slijepcevic, M. Potkonjak, Power efficient organization of wireless sensor networks, in: Proceedings of the IEEE International Conference on Communications (ICC), Helsinki, Finland, USA, June11–14, 2001.
[31] H.W. Tsai, C.P. Chu, T.S. Chen, Mobile object tracking in wireless sensor networks, Comput. Commun. 30 (18) (2007) 11–1825.
[32] P. Vicaire, T. He, Q. Cao, T. Yan, G. Zhou, L. Gu, L. Luo, R. Stoleru, A.J. Stankovic, T.F. Abdelzaher, Achieving long-term surveillance in vigilnet, ACM Trans. Sensor Networks 5 (1) (2009) 1–39.
[33] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, in: Proceedings of ACM International Conference on Embedded Networked Sensor Systems (SenSys), Los Angeles, California, USA, November 5–7, 2003.
[34] X. Wang, M. Fu, H. Zhang, Target tracking in wireless sensor networks based on the combination of KF and MLE using distance measurements, IEEE Trans. Mob. Comput. 11 (4) (2012) 567–576.
[35] Q.X. Wang, W.P. Chen, R. Zheng, K. Lee, L. Sha, Acoustic target tracking using tiny wireless sensor devices, in: Proceedings of the International Workshop on Information Processing in Sensor Networks (IPSN), Palo Alto, California, USA, April 22–23, 2003.
[36] X. Wang, J. Ma, S. Wang, D. Bi, Prediction-based dynamic energy management in wireless sensor networks, in: Proceedings of the IEEE Sensors, Atlanta, Georgia, USA, October 28–31, vol. 7, 2007, pp. 251–266.
[37] H. Yang, B. Sikdar, A protocol for tracking mobile targets using sensor networks, in: Proceedings of the IEEE International Workshop on Sensor Networks Protocols and Applications, Anchorage, Alaska, USA, May 11, 2003.
[38] F. Ye, G. Zhong, S. Lu, L. Zhang, PEAS: a robust energy conserving protocol for long-lived sensor networks, in: Proceedings of the International Conference on Distributed Computing Systems (ICDCS'03), Providence, RI, USA, May 19–22, 2003.
[39] S.B. Young, Evaluation of pedestrian walking speeds in airport terminals, J. Trans. Res. Board 1674 (1999) 20–26.
[40] W. Zhang, G. Cao, DCTC: dynamic convoy tree-based collaboration for target tracking in sensor networks, IEEE Trans. Wireless Commun. 3 (5) (2004) 1689–1701.
[41] W. Zhang, G. Cao, Optimizing tree reconfiguration for mobile target tracking in sensor networks, in: Proceedings of the 23rd Conference of the IEEE Communications Society, Hong Kong, China, March 7–11, 2004.
[42] F. Zhao, J. Shin, J. Reich, Information-driven dynamic sensor collaboration for target tracking, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2002, Orlando, Florida, USA, May 13–17, 2002.