Following Targets for Mobile Tracking in Wireless Sensor Networks

TIAN WANG, Huaqiao University; The Hong Kong Polytechnic University
ZHEN PENG, Huaqiao University
JUNBIN LIANG, Guangxi University
SHENG WEN, Deakin University
MD ZAKIRUL ALAM BHUIYAN, Temple University
YIQIAO CAI, Huaqiao University
JIANNONG CAO, The Hong Kong Polytechnic University

Traditional tracking solutions in wireless sensor networks based on fixed sensors have several critical problems. First, due to the mobility of targets, a lot of sensors have to keep being active to track targets in all potential directions, which causes excessive energy consumption. Second, when there are holes in the deployment area, targets may fail to be detected when moving into holes. Third, when targets stay at certain positions for a long time, sensors surrounding them have to suffer heavier work pressure than do others, which leads to a bottleneck for the entire network. To solve these problems, a few of mobile sensors are introduced to follow targets directly for tracking because the energy capacity of mobile sensors is less constrained and they can detect targets closely with high tracking quality. Based on a realistic detection model, a solution of scheduling mobile sensors as well as fixed sensors for target tracking is proposed. Moreover, the movement path of mobile sensors has provable performance bound compared with the optimal solution. Results of extensive simulations show that mobile sensors can improve the tracking quality even if holes exist in the area and can reduce energy consumption of sensors effectively.

CCS Concepts:•**Networks** → **Sensor networks;**

Additional Key Words and Phrases: Wireless sensor networks, target tracking, mobile nodes

ACM Reference Format:
Tian Wang, Zhen Peng, Junbin Liang, Sheng Wen, Md Zakirul Alam Bhuiyan, Yiqiao Cai, and Jiannong Cao, 2015. Following mobile targets for tracking in wireless sensor networks *ACM Trans. Sensor Netw.* V, N, Article A (YYYY), 24 pages.
DOI: 0000001.0000001

1. INTRODUCTION

Wireless sensor networks (WSNs) generally consist of numerous inexpensive sensors, which are powered by portable energy supplies and can sense physical events in order to collect environmental information [Akyildiz et al. 2002]. Once deployed, the entire network usually needs to work for a long time without intervention. One of the most

This work is supported by the National Natural Science Foundation (NSF) of China under grant 61572206 and 61305085, the Natural Science Foundation of Fujian Province of China under grant 2014J01240, and the Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University (No.ZQN-PY308).
Author's addresses: T. Wang, Z. Peng, and Y. Cai, College of Computer Science & Technology, Huaqiao University, Xiamen, Fujian, China; J. Liang, School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, China; S. Wen, School of Information Technology, Deakin University, Melbourne Burwood, Victoria, Australia; M. Z. A. Bhuiyan, Department of Computer and Information Sciences, Temple University, Philadelphia, PA, US; J. Cao, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong.

DOI: 0000001.0000001

typical applications in WSNs is mobile target tracking, whose goal is to detect and track mobile targets, such as illegal intruders and wildlives. Differed from discrete events detection [Katenka et al. 2008; Amaldi et al. 2012], target tracking requires monitoring targets continuously, namely, monitoring targets during their mobility processes.

However, there are several problems for traditional WSNs to track mobile targets. First, sensors in networks are powered by low-cost batteries and should work for a long time in an unattended manner, so their energy is extremely limited and cannot be recharged [Akyildiz et al. 2002]. When the target moves, a lot of sensors in the network have to keep being active in order to track targets in all potential directions, which causes excessive energy consumption. Second, in practice, it is difficult to deploy sensors evenly in the deployment area. Generally, sensors are scattered randomly so coverage holes may exist in networks [Nordio and Chiasserini 2011]. If targets enter the area of holes, no sensors can monitor them effectively. Third, when targets stay at certain positions for a long time, sensors surrounding them have to undertake heavier work than do other sensors, which causes excessive energy consumption and leads to a bottleneck for the entire network. As a result, it may generate coverage holes and decrease the lifetime of the network.

To deal with the problems mentioned above, this paper introduces a few of mobile sensors into traditional WSNs for target tracking, which is called *mobile tracking*. In general, mobile sensors can contribute to target tracking with a couple of advantages. On the one hand, mobile sensors can move towards targets for persistent monitoring during the entire tracking process. As mobile sensors have the ability to move close to targets, their monitoring is more effective than that of fixed sensors. A few of mobile sensors may finish the work used to be done by a lot of fixed sensors which can then turn themselves into the sniff state thus much energy can be saved. The energy capacity of mobile sensors is less constrained as they can replenish their energy because of their mobility [Lembke et al. 2011; Xing et al. 2012]. On the other hand, when there are coverage holes in the deployment area, mobile sensors can still move into these areas and detect targets. The negative effects of coverage holes can be greatly eliminated. To summarize, Table I compares the characteristics of traditional fixed tracking (all-static) methods and mobile tracking (with a few of mobile sensors) methods in several metrics.

Table I: Comparisons of fixed and mobile tracking methods

Performance Metrics	Fixed Tracking	Mobile Tracking
Energy Consumption	High (non-replenishable)	Low (replenishable)
Detection Quality	Low	High
Fault Tolerance	High	Low
With Coverage Holes	Out of service	Still work

Therefore, we propose a tracking solution based on a hybrid network that consists of a few of mobile sensors and a large number of fixed sensors. Moreover, we exploit mobile sensors to follow targets directly in order to detect them in short distance. In this way, the tracking quality is guaranteed and the energy consumption of fixed sensors can be significantly reduced as much fewer of them need to be in the active state. However, there are several challenges for utilizing mobile sensors in directly tracking targets. First, as the velocity of mobile sensors is limited in practice [Lembke et al. 2011], their movement should be scheduled reasonably. Hence, mobile sensors should be selected appropriately to conduct tracking and cooperate effectively with fixed sensors. Second, the efficiency of collaboration between mobile sensors and fixed sensors is very important. The introduction of mobile sensors should not cause heavy network communication between two kinds of sensors. Meanwhile, fixed sensors should be well scheduled to be active for target tracking or turned off for energy saving.

In this paper, we aim to utilize advantages of mobile sensors and address these challenges at the same time. The main contributions of this paper are listed as follows:

(1) As far as we know, we are the first to make mobile sensors move to follow the target directly for detection in cooperation with fixed sensors. A novel problem—continuous tracking problem based on a realistic model is proposed.
(2) We design an approximation solution for scheduling mobile sensors to cooperate with fixed sensors, which guarantees the quality of tracking and reduces the energy consumption. The movement paths of mobile sensors have provable performance bound compared with the optimal solution.
(3) We extend the solution for tracking multiple targets. Based on a action force mechanism, mobile sensors can be almost evenly distributed to track every single target, which improves the whole tracking quality.
(4) We conduct extensive simulations to compare the proposed solution with traditional solutions, and the effectiveness is validated by results.

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 introduces the system model, including the detection model and network model. The problem of continuous tracking is defined in Section 4. Section 5 presents the solution for tracking a single target and Section 6 presents the extension solution for tracking multiple targets. Simulation results are demonstrated in Section 7, and Section 8 concludes this paper.

2. RELATED WORK

Target tracking is an important application in WSNs, which attracts a large number of scholars to conduct research [Naderan et al. 2012; Demigha et al. 2013]. Most nodes in traditional WSNs are deployed fixedly thus without mobile abilities. As fixed nodes cannot follow the moving target, many nodes are generally required to be in the active state which leads to huge amount of energy consumption [Tsai et al. 2007; Chen et al. 2004]. Energy efficiency is one of the most significant problems to be considered in WSNs, so reducing the number of active nodes is a way to conquer it. For instance, Bhuiyan et al. [Bhuiyan et al. 2010] proposed a prediction-based target tracking protocol. They tried to reduce energy consumption of nodes by beforehand activating nodes near the predicted position and switching other unnecessary nodes into sleep mode, which was based on predicting the trajectory of the target by its moving velocity and direction. Teng et al. [Teng et al. 2012] proposed two signal processing oriented cluster management strategies, the proactive and reactive cluster management, to deal with energy and longevity constraints. They designed a Dijkstra-like algorithm to form active cluster based on the relation between predictive target distributions and candidate nodes. However, tracking faults caused by coverage holes may weaken the effectiveness of this kind of solutions, and mobile nodes can alleviate the defect to some extent. Recent years, as energy of mobile nodes is unrestricted compared to ordinary fixed nodes, researchers began to introduce mobile nodes into WSNs [Martínez and Bullo 2006; Zou and Chakrabarty 2007; Yang et al. 2014; Shan and Tan 2005; Kumar and Parvin 2013; Mourad et al. 2012; Bai et al. 2012; Hwang et al. 2008; Wimalajeewa and Jayaweera 2010; Tan et al. 2010].

The research conducted by Martínez et al. [Martínez and Bullo 2006] was based on mobile sensor networks. They designed distributed motion coordination algorithms that increase the information gathered by a network in static and dynamic target-tracking scenarios. Those mobile sensors are steered to an optimal deployment and are amenable to a decentralized implementation. Their work focuses on decentralized filters and data-fusing methods for estimation during the tracking, which is only suitable for mobile sensors. Zou et al. [Zou and Chakrabarty 2007] described a distributed mobility management scheme for mobile sensor networks. The proposed scheme considered node movement decisions as part of a distributed optimization problem with readings of the network's situation. The experiment results denoted that the mobile network performs better than the static network in tracking accuracy. Zhou et al. [Zhou and Roumeliotis 2008] studied the problem of tracking trajectory generation for a team of mobile sensors tracking a moving target using distance-only measurements. They proposed algorithms for determining the set of feasible locations where each sensor should move in order to collect the most informative measurements. A mathematical model for single target tracking was formulated by Hu et al. [Hu and Hu 2010] using mobile nonlinear scalar range sensors. They also proposed a sensor deployment strategy for mobile sensors and a nonlinear convergent filter for estimating the trajectory of the target. However, in these literatures, they assumed that all of the sensors are capable of mobility, which is expensive for real applications. In our network, we use a hybrid network including mobile nodes and fixed nodes together.

In [Shan and Tan 2005], Shan et al. presented a mobile sensor deployment algorithm within a hybrid sensor network, which consists of a few mobile sensors and a relatively large number of static sensors. In this scalable algorithm, fixed sensors in the network construct the cluster for a target. Meanwhile, those mobile sensors are used to fill the coverage holes inside the cluster. Hwang et al. [Hwang et al. 2008] proposed a solution for target tracking with mobile sinks. The sink can move freely in networks and send the tracking request to nodes. The mobile sink can collect the tracking information of the target, which is detected by fixed nodes. Wimalajeewa et al. [Wimalajeewa and Jayaweera 2010] proposed a new mobility assisted tracking (MAT) algorithm to track a single target in a hybrid sensor network consisting of both fixed and mobile nodes. The network is assumed to be partitioned into clusters and cluster heads are formed from a set of high capacity static nodes. They exploited the node mobility in the network to dynamically maintain a certain coverage level at the predicted target location. That is, those mobile nodes are moving towards the position which is not covered to the desired coverage level by static nodes. Kumar et al. [Kumar and Parvin 2013] dealt with a single target tracking scenario performed with static as well as mobility optimized mobile nodes in an energy conserving manner. The mobile nodes in this designed hybrid sensor network for target tracking are taken advantage of to improve the target position estimation by swarm intelligence optimization. In these literatures, mobile nodes play an important role in the tracking process. However, they are generally used to improve the performance of the network in a certain aspect rather than track targets directly. In our proposed algorithm, we exploit mobile nodes to follow targets directly, as well as collaborate with fixed nodes.

Mourad et al. [Mourad et al. 2012] addressed the problem of single target tracking in controlled mobility sensor networks. They proposed a strategy for managing sensors mobility, which consists of estimating the current position of a single target as well as predicting its following location. Similarly, Bai et al. [Bai et al. 2012] proposed a coordinative moving strategy for autonomous mobile sensor networks to guarantee the target can be detected in each observed step. In the scheme, the current position of the target is used to predict its next time-step position. Yang et al. [Yang et al. 2014] considered a tracking scenario mixed with both additive and multiplicative noises in measurements. They proposed a coordination strategy, including sensor selection and sensor motion, to improve the tracking accuracy. In these literatures, they focus on the selection and motion strategy of sensors, rather than the energy efficiency. In our proposed solution, the energy consumption of fixed nodes is a major consideration.

In conclusion, There are several problems in existing tracking methods with both fixed sensors and mobile sensors. First, mobile sensors in these literatures are usually used to improve the performance of the network, such as sensing coverage in [Shan and Tan 2005; Wimalajeewa and Jayaweera 2010]. However, the use of mobile sensors in these methods may not help to improve the tracking quality directly or effectively. Second, in these methods, mobile sensors and fixed sensors have little cooperation in tracking process. Most literatures only concentrate on how to manage mobile sensors such as in [Mourad et al. 2012; Yang et al. 2014]. Third, few methods consider the scenario with multiple targets. In this paper, we exploit mobile sensors to follow and track targets directly. During the process, fixed sensors and mobile sensors cooperate together to ensure that the target has been detected effectively. In this way, we try to keep the tracking quality no less than a given threshold. Moreover, we extend our method for tracking multiple targets and try to ensure the average tracking performance for every target.

3. PRELIMINARIES

Now we introduce the detection model in Section 3.1 and the network model in Section 3.2. Based on these realistic models, the definition of tracking probability is presented in Section 3.3.

3.1. Detection Model

We use the detection model based on the detection probability similar to literature [Tan et al. 2010; Sheng and Hu 2005; Niu et al. 2004]. The target itself emits a signal, e.g., acoustic signal. A node can detect the target by measuring the emitting energy. Meanwhile, the measurements of nodes are contaminated by background noise which is modeled as additive Gaussian noise. Therefore, when the target appears, the energy e_i measured by node i is mixed by the signal energy $e_s(d_i)$ and the noise energy e_n, so e_i is given by

$$e_i = e_s(d_i) + e_n \tag{1}$$

where the d_i in the signal energy $e_s(d_i)$ is the Euclidean distance between node i and the target. The $e_s(d_i)$ attenuates with the increasing of d_i and can be expressed as

$$e_s(d_i) = \begin{cases} \dfrac{S_0}{(d_i/d_0)^k} & if\ d_i > d_0 \\ S_0 & if\ d_i \leq d_0 \end{cases} \tag{2}$$

where d_0 is a constant as a reference factor, and S_0 is the signal energy measured within the distance d_0 to the source. k is an attenuation factor which is from 2 to 5. The noise energy e_n approximately satisfies the Gaussian distribution with mean equal to μ and variance equal to σ^2. Therefore, the total signal energy value measured by node i also follows a Gaussian distribution

$$e_i \sim \mathcal{N}(\mu + e_s(d_i), \sigma^2)\ . \tag{3}$$

3.2. Network and Communication Models

The network consists of a large number of fixed nodes and a few of mobile nodes. A small number of mobile nodes will not bring excessive cost. They can move anywhere in a random walk way [Shah et al. 2003]. The communication model of nodes is the unit-disk graph model [Zhang et al. 2014; Tuna et al. 2014; Ruhrup and Stojmenovic 2013]. In this model, nodes have a certain distance for communication, which is called communication radius. If the distance between two nodes is no larger than the communications radius, they can exchange information with each other. Nodes in the network can communicate with each other through ad hoc networks. Nodes know their own positions by Global Positioning System (GPS) or other localization methods such as [Chu and Jan 2005; Stoleru et al. 2007]. When the target is being detected, its positions can be calculated by nodes according to existing localization methods such as [Chen et al. 2002]. Fixed nodes can switch themselves among active, sleep and sniff states in order to save energy. That's to say, a node is in the active state when it undertakes the tracking work, and is in the sniff state when stopping working for saving energy [Vicaire et al. 2009; Atia et al. 2011]. When nodes are in the sniff state, they stay in the sleep state and wake up for a relatively short period periodically, during which they can detect whether the target appears and locate it. The energy consumption of mobile nodes is less constrained as they can replenish their energy because of the mobility [Lembke et al. 2011], so we only consider the energy consumption of fixed nodes.

According to the measured value e_i mentioned in Section 3.1, an individual node compares it with a threshold λ. If the value is not less than λ, the node can detect and track the target, otherwise it cannot. This process is called local decision. The nodes surrounding the target collect all local decisions and make a final determination [Varshney 1996].

3.3. Tracking Probability Definition

According to Formulas (2) and (3), when the target appears, the probability of getting the measurement e_i for node i is expressed as

$$p(e_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(e_i - \mu - e_s(d_i))^2}{2\sigma^2}\right) . \tag{4}$$

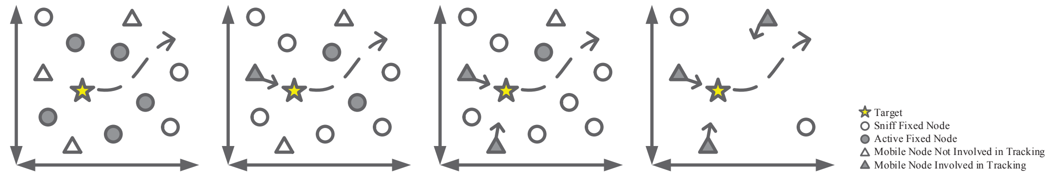
Node i compares its measurement with a threshold λ. If its measurement is greater than λ, it makes decision 1 which means it can detect the target, otherwise it makes decision 0. This decision rule is called Likelihood Ratio Test [Tan et al. 2010]. Therefore, the detection probability P_D^i of individual node i is expressed as

$$P_D^i = \int_{\lambda}^{\infty} p(e_i)\mathrm{d}e_i . \tag{5}$$

When n nodes track the same target, the tracking probability is given by

$$P_D = 1 - \prod_{i-1}^{n}(1 - P_D^i) . \tag{6}$$

And P_D can be defined as the tracking quality. According to Formula (6), we can find more nodes tracking the target will lead to a higher tracking probability. However, the increasing of involved fixed nodes causes more energy consumption. Hence the tracking task should be accomplished by mobile nodes in the network as much as possible. Moreover, according to Formula (5), nodes closer to the target lead to higher



(a) 5 fixed sensors are required to track. (b) 1 mobile sensor involved then only 2 fixed sensors are required. (c) 2 mobile sensors involved and 1 fixed sensor is required. (d) 3 mobile sensors involved and no fixed sensors are required.

Fig. 1: A numerical example of continuous tracking with mobile sensors. Mobile sensors can move close to targets, improving the detection probability and saving the energy.

tracking probabilities, which gives us the inspiration that mobile nodes should move close to the target as much as possible in order to improve the detection accuracy.

4. TARGET TRACKING WITH MOBILE SENSORS

Based on preliminaries defined above, we introduce the problem of continuous tracking. The problem description is given in Section 4.1. An numerical example is illustrated in Section 4.2.

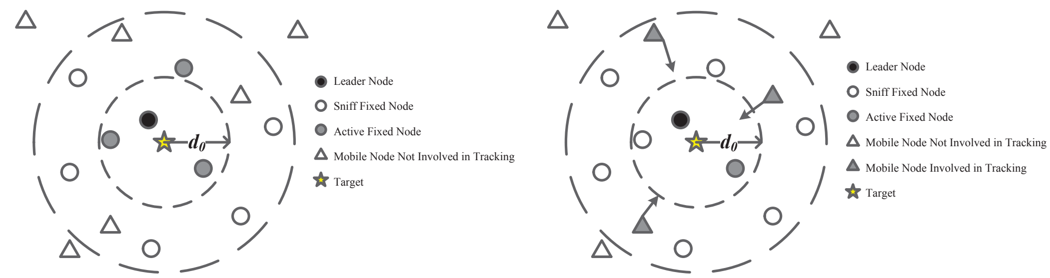
4.1. Problem Description

There are s fixed sensors and m mobile sensors randomly deployed in a two-dimension $L \times L$ plane surveillance area, $s \gg m$. Suppose that the velocity of mobile sensors is v_m and the maximum velocity of the target is v_t ($v_m < v_t$). Targets move in a random walk way [Shah et al. 2003]. Once the target is detected, mobile sensors and fixed sensors cooperatively track it. During the tracking process, the tracking probability P_D of the target should be not less than a given threshold α, which represents the tracking quality. The time that the tracking quality is satisfied is called effective monitoring time. Therefore, the goal is to schedule both fixed sensors and mobile sensors to guarantee the tracking probability and decrease the energy consumption of fixed sensors as much as possible.

4.2. Numerical Example

A numerical example is illustrated in Figure 1, where circles, triangles and a star stand for fixed sensors, mobile sensors and the target, respectively. These solid nodes represent active sensors and the hollow ones represent sleeping sensors. Suppose that the requirement of tracking probability is 80%. The example contains four different scenarios: a) when no mobile sensors participates in tracking, 5 fixed sensors need to be active in order to meet the requirement 80%; b) when one mobile sensor tracks the target, as this mobile sensor has the ability to move close to the target and detect it, the detection accuracy is improved and only 2 fixed sensors need to be active; c) when the number of mobile sensors tracking the target is up to 2, to meet the requirement only 1 fixed sensor is needed; d) when the target moves to some areas without fixed sensors (for example, areas with holes), 3 mobile sensors keep detecting the target to meet the tracking requirement.

As can be seen in this example, using mobile sensors significantly reduce the number of fixed sensors involved in the process of tracking with the same continuous tracking



(a) Some nearest fixed nodes are in active mode to meet the requirement of P_D.

(b) Mobile nodes move towards the target and detect it. If the detection probability can be satisfied, some fixed nodes can be switched to the sniff state.

Fig. 2: An illustration of mobile tracking solution.

requirement. As a result, the energy consumption of fixed sensor can be saved and the lifetime of networks can be prolonged. Therefore, our focus is on how to efficiently schedule mobile sensors to take charge of the tracking work.

5. SINGLE TARGET TRACKING SOLUTION

According to mobility planning, we propose our algorithm for tracking one single target. Later on, we extend this algorithm for tracking multiple targets in Section 6. The algorithm for mobile tracking is shown in Section 5.1 and 5.2, and performance analyses are presented in Section 5.3.

5.1. Mobile Tracking Algorithm

This section proposes the tracking solution called *MTTA* (Mobile Tracking based on Tacit Agreement) solution. After deployment, fixed nodes acquire the location information of their neighbor nodes by exchanging beacon messages. This information is used for selecting nodes involved in tracking. Nodes can detect the presence of targets by traditional methods [Amaldi et al. 2012]. Nodes also know their own positions and the target's position by existing localization methods [Chu and Jan 2005; Chen et al. 2002]. At first, nodes exchange their position information with each other, and the nearest node from the target declares itself to be the first L_0, i.e. leader node, which is responsible for the local scheduling task. According to the current location of the target, L_0 selects i nearest node (set S in Algorithm 1) to the target (including itself) in order to detect it and satisfy the condition that $P_D \geq \alpha$, just like solid nodes in Figure 2a. As L_0 knows locations of the target and nodes, this selection process needs no communications.

After that, L_0 informs these i nodes to be active for tracking the target. For these nodes, if a node detects that the distance from L_0 to the target is greater than $2d_0$ (how to detect will be represented in Section 5.2), or its two consecutive detection values are both less than $\mu + 3\sigma$, it can autonomously switch to the sniff state for energy saving as the target has moved away. This kind of autonomous action is called a Tacit Agreement. The advantage of the tacit agreement is that L_0 needs no communication with these detecting nodes when the target moves away from them. If without the tacit agreement, L_0 needs to inform those detecting nodes to switch into the sniff state, which will increase the communication cost. Therefore, the tacit agreement can decrease necessary communication cost. Figure 3 shows states transition of nodes. Once the distance between L_0 and the target is greater than $2d_0$, a new L_0 is chosen as the

Algorithm 1 Mobile Tracking based on Tacit Agreement (*MTTA*)

Fixed nodes collect location information of the neighbor nodes and detect the appearance of the target;
The nearest node from the target is chosen as L_0, which is a local coordinator;
if the distance between L_0 and the target $> 2d_0$ **then**
    The nearest node from the target becomes the new L_0 and gets information from the old one;
end if
L_0 chooses some of the nearest nodes from the target so that $P_D > \alpha$ and adds them into set S;
L_0 informs the nodes in S to be active for tracking the target;
for node $i \in S$ **do**
    if two consecutive detection values are less than $\mu + 3\sigma$ $\|$ the distance from the target to L_0 is greater than $2d_0$ **then**
        i switches to the sniff state and is removed from S;
    end if
end for
L_0 informs the mobile nodes within $2d_0$ distance of the target to move towards the current location of the target;
Mobile nodes within $2d_0$ distance of the target are informed the target location every t_0 time, which is the work cycle;
Mobile nodes whose distance to the target is greater than $2d_0$ are informed the target location every t_i time, where $t_i = 2d_0/(v_m + v_t)$;
loop
    //During the mobility process:
    if $P_D > \alpha$ after stopping the farthest monitoring node r_j to the target from detecting **then**
        r_j switches to the sniff state and is removed from S;
        if r_j is L_0 **then**
            The nearest node from the target becomes the new L_0 and gets information from the old one;
        end if
    end if
    if $P_D < \alpha$ **then**
        Add the nearest node r_k which is $\notin S$ to S and switch it to the active state;
    end if
end loop

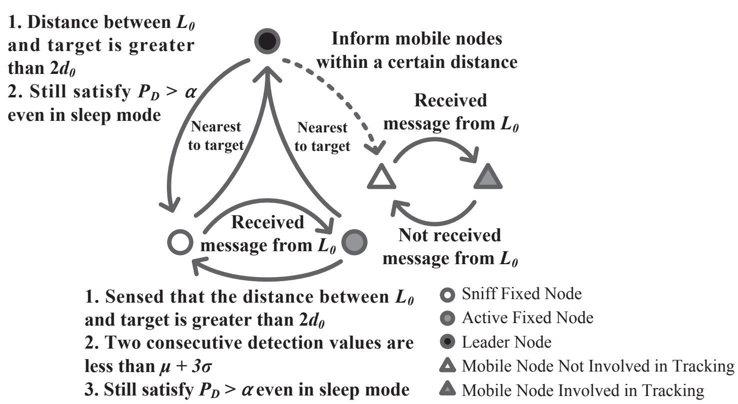


Fig. 3: An illustration of conditions for nodes' states transition.

nearest node to the target and the former L_0 transfers related information of nodes and the target to the new one, such as the location of the target. According to this information, the new L_0 takes over the local scheduling task.

More importantly, L_0 informs mobile nodes within certain distance from the target to involve in tracking and move to the target. The certain distance can be adjusted according to practical situations. Solution details can be found in Algorithm 1. Figure 2b illustrates this solution. Mobile nodes within the larger dashed circle around the star, e.g. the target, are informed to move towards the target. The initial locations of these moving mobile nodes are recorded. As shown in Figure 4, during their moving process, L_0 predicts the current distance between mobile nodes and the target according to recorded initial locations of mobile nodes (since the velocity of mobile node is a

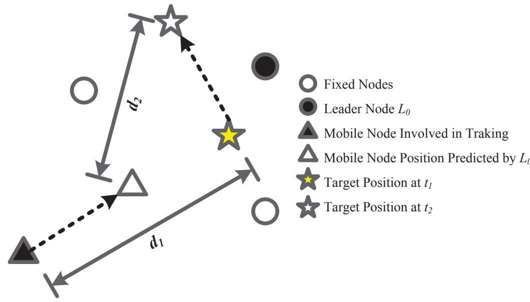


Fig. 4: L_0 predicts the position of mobile nodes and the target to calculate P_D.

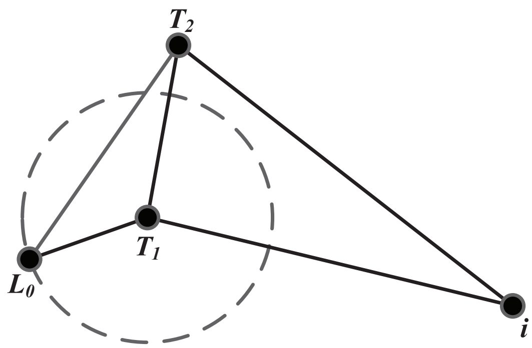


Fig. 5: An illustration of the positional relationship among the target, L_0 and node i.

constant), hence the tracking probability P_D of the target is calculated without communication with mobile nodes. If P_D is greater than α and after stopping the current furthest monitoring node (denoted as r_j, $1 \le j \le i$) to the target from detecting, the P_D is still greater than α, r_j will switch to the sniff state for energy saving. If node r_j happens to be L_0, the nearest mobile node from the target will become the new L_0. On the contrary, if P_D is less than α, it represents that the number of tracking nodes is not enough, and the nearest node (denoted as r_k, $1 \le k \le i$) which is not involved in tracking should switch to be active and start detecting.

5.2. Node Predicts the Distance Between L_0 and the Target

According to the difference between two signal measurements, which are the initial measurement and the current measurement respectively, node i can determine whether the distance from the target to L_0 has increased to more than $2d_0$. As showed in Figure 5, the target's initial location is T_1 and its current location is T_2. The locations of L_0 and node i are illustrated in the figure. Suppose that the distance between node i and the target is $T_i i = d$. According to Formula (2), the difference between two signal measurements of node i can help the node determine whether the distance from the node to the target has increased by $4d_0$, i.e., $T_2 i = T_1 i + 4d_0$. According to the relationship between the sides of a triangle, the distance between T_1 and T_2 satisfies $T_1T_2 > 4d_0$. Meanwhile, we have $L_0T_1 + L_0T_2 > T_1T_2$, i.e., $L_0T_2 > T_1T_2 - L_0T_1$, and $L_0T_1 < 2d_0$, then the distance between L_0 and T_2 satisfies $L_0T_2 > 2d_0$.

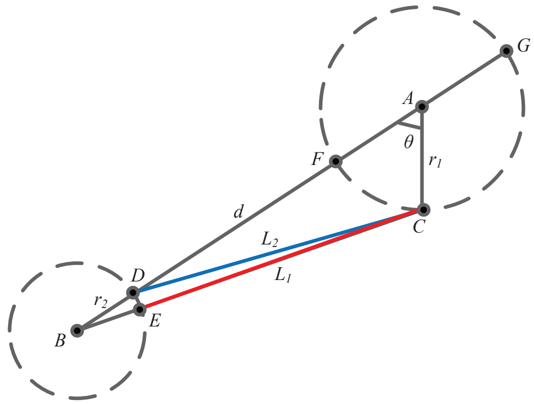


Fig. 6: An illustration of the positional relationship between the target at A and the node at B.

5.3. Algorithm Analysis

In the following, we show analysis of our algorithm. The algorithm name—"Tacit Agreement" is so named because we hope the communicating parties have a tacit agreement to decrease the communication cost. We have the following theorems:

THEOREM 5.1. *The probability that a fixed node should have been in the sniff state but stays in the active state is a rare event.*

PROOF. Suppose that fixed node i is informed by L_0 to participate in tracking, then there are two situations in which node i no longer needs to keep tracking. In the first situation, due to the mobility of the target, a new L_0 is chosen, but it doesn't inform node i to participate tracking. According to Algorithm 1, the new L_0 is responsible for informing the nodes which need to detect targets. As node i is not informed by the new L_0, it will switch to the sniff state according to the tacit agreement proposed in Section 5.1. In the second situation, the target has not actually existed and the tracking notice is a misjudgment. Since the detecting probability for targets is not one hundred percent, this situation happens possibly. In this situation, only when two consecutive detection values are both greater than $\mu + 3\sigma$, node i needs to keep active. According to the PauTa criterion of Gaussian distribution, this probability is about $(1-0.99865)^2 \approx 1.8\text{E-}6 < 0.05$. Generally, the event whose happening probability is less than 0.05 is called a rare event. Therefore, the theorem is proved. □

THEOREM 5.2. *The event that a fixed node, which was within $3d_0$ distance of the target, should have been in the active state but is in the sniff state is a rare event.*

PROOF. A node, which was within $3d_0$ distance of the target, turns into the sniff state only in two situations. One is that L_0 has not chosen it to be active. In this situation, this node is unnecessary to participate in tracking. The second situation is that the node has been pointed out by L_0 but it turns into the sniff state later on, under the condition that two consecutive detection values are less than $\mu + 3\sigma$. Since the detection value of the node satisfies the Gaussian distribution with mean equal to $\mu + e_s(d_i) = \mu + S_0/9$ and variance equal to σ^2, the probability of detecting value being less than $\mu + 3\sigma$ is expressed as:

$$\mathrm{P}(X < \mu + 3\sigma) = \Phi(\frac{\mu + 3\sigma - \mu - S_0/9}{\sigma}) = \Phi(3 - \frac{S_0}{9\sigma}) \quad (7)$$

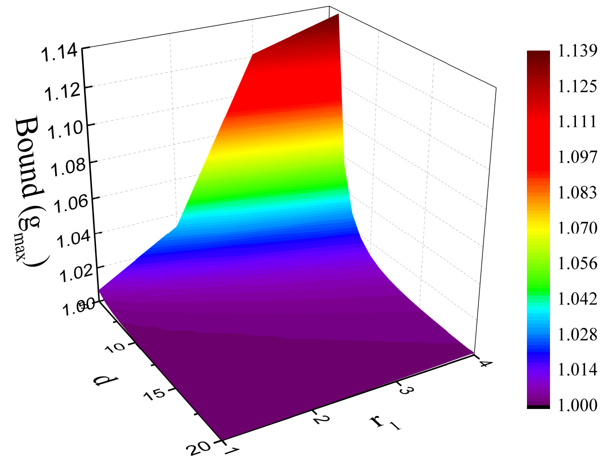


Fig. 7: The numerical result of the bound under different parameters.

S_0 and σ are both constants, and generally $S_0 \gg \sigma$. According to literature [Tan et al. 2010], $S_0/\sigma > 36$. Substituting it to Formula (7) and we get $\mathrm{P}(X < \mu + 3\sigma) < \Phi(-1) \approx 0.16$. Therefore, the probability that two consecutive detecting values are both less than $\mu + 3\sigma$ is 0.16^2, which is less than 0.05. Therefore, it's a rare event and the theorem is proved. $\square$

These two theorems show that our algorithm can work well without frequent and explicit communications. In the next, we give the analysis of performance bound. We have the following theorem regarding the approximation ratio of *MTTA*.

THEOREM 5.3. *While the location of the target is noticed every t seconds, the approximation ratio of the MTTA algorithm comparing with the optimal solution is no greater than g_{max}, which is a constant.*

PROOF. As the Figure 6 shows, A and B are initial locations of a target and a mobile node, respectively. Suppose the current time is 0. According to our algorithm, the mobile node will move towards A, along the line BA and t_i seconds later it will be at location D, moving the distance $r_2 = v_t \times t_i$. Without loss of generality, t_i time later, suppose the target is at location C and the movement distance is $r_1 = v_m \times t_i$. The length of DC (denoted as L_2) is the distance from the mobile node to the target. Obviously, the optimal movement path is BC and the mobile node should be at location E at time t_i. The length of EC (denoted as L_1) is the optimal distance to the target. If the distance from mobile node to the target is no greater than $2d_0$, i.e. $|AB| \leq 2d_0$, the movement path of mobile node is the optimal since the latest location of the target is known. If the distance from mobile node to the target is greater than $2d_0$, then $|AB| > 2d_0 = (v_m + v_t) \times t_i = r_1 + r_2$ and $|BC| > r_1$.

Thus, according to the law of cosines, we have

$$L_1 = \sqrt{r_1^2 + d^2 - 2r_1 d\cos\theta} - r_2$$
$$L_2 = \sqrt{r_1^2 + (d - r_2)^2 - 2r_1(d - r_2)\cos\theta}\ .$$

Let

$$g(\theta) = \frac{L_2}{L_1} = \frac{\sqrt{r_1^2 + (d-r_2)^2 - 2r_1(d-r_2)\cos\theta}}{\sqrt{r_1^2 + d^2 - 2r_1 d\cos\theta} - r_2} .$$

So,

$$g'(\theta) = \frac{\mathrm{d}g}{\mathrm{d}\theta} = \frac{\frac{(\sqrt{r_1^2+d^2-2r_1 d\cos\theta}-r_2)r_1(d-r_2)\sin\theta}{\sqrt{r_1^2+(d-r_2)^2-2r_1(d-r_2)\cos\theta}} - \frac{\sqrt{r_1^2+(d-r_2)^2-2r_1(d-r_2)\cos\theta}\,r_1 d\sin\theta}{\sqrt{r_1^2+d^2-2r_1 d\cos\theta}}}{(\sqrt{r_1^2 + d^2 - 2r_1 d\cos\theta} - r_2)^2} .$$

Let $g'(\theta) = 0$ and get

$$\frac{(\sqrt{r_1^2 + d^2 - 2r_1 d\cos\theta} - r_2)r_1(d-r_2)\sin\theta}{\sqrt{r_1^2 + (d-r_2)^2 - 2r_1(d-r_2)\cos\theta}} = \frac{\sqrt{r_1^2 + (d-r_2)^2 - 2r_1(d-r_2)\cos\theta}\,r_1 d\sin\theta}{\sqrt{r_1^2 + d^2 - 2r_1 d\cos\theta}} .$$

And we have

$$\cos\theta = \frac{[(d-r_2)^2 + 2(d-r_2)d - r_1^2]r_1}{2(d-r_2)^2 d} .$$

In this situation, $g(\theta)$ achieves its maximum that

$$g_{max} = \frac{\sqrt{r_1^2 + (d-r_2)^2 - \frac{[(d-r_2)^2+2(d-r_2)d-r_1^2]r_1^2}{(d-r_2)d}}}{\sqrt{r_1^2 + d^2 - \frac{[(d-r_2)^2+2(d-r_2)d-r_1^2]r_1^2}{(d-r_2)^2}} - r_2} .$$

□

Figure 7 shows numerical result of the performance bound with $r_2 = 1$. Different colors represent different sizes of the bound values. We find that the parameter d is the most relative one to the bound value. The bound value g_{max} is in inverse proportion to the value of d, which is the distance between the target and mobile nodes. The value of r_1 has little effect on the result of g_{max}. This means that with the increasing of target velocity, the performance of our algorithm declines a little. Taken as a whole, however, the performance bound is acceptable.

6. MULTIPLE TARGETS TRACKING

Up to now, we have discussed the situation of tracking single target. In the following, we introduce the multiple targets tracking solution. First, the problem of multiple targets is described in Section 6.1. Then, an action force mothed is introduced in Section 6.2. Based on it, the multiple targets tracking algorithm is designed in Section 6.3.

6.1. Problem Description

The basic deployment of multiple targets tracking scenario is similar to the single target one. There are s fixed sensors and m mobile sensors in a two-dimension $L \times L$ plane surveillance area. Now, Θ targets randomly move or stay in the area, and $\Theta > 1$. For multiple targets, we need to ensure the entire tracking quality of them rather than that of a single target. Suppose P_{Dk} is the tracking probability for target k, and $k = 1, 2, \ldots, \Theta$. Therefore, the goal is to ensure the tracking probability of every target and to decrease the energy consumption of fixed sensors as much as possible.

For multiple targets tracking, the problem is how to guarantee the tracking quality of all targets. There is a situation that, for some reasons, some targets may get too

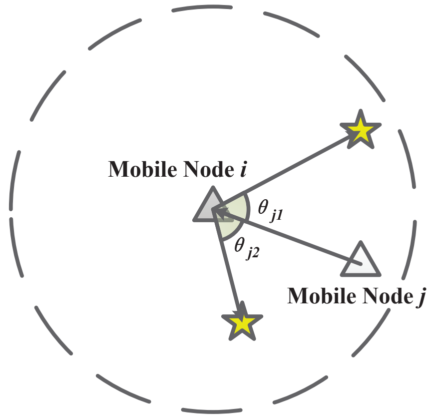


Fig. 8: An illustration of attractive force and repulsive force.

much attention. That is, most of mobile sensors are following them while other targets are getting ignored. This can be called an inequitable situation. It is obvious that if an inequitable situation happens, the tracking probability for those neglected targets will suffer. Therefore, in order to avoid this kind of situation, mobile sensors should rationally decide which target is to be tracked based on some strategies, so the mobile sensors participating tracking are evenly distributed. Meanwhile, the specific strategy is supposed to be 1) effective to choose an appropriate target, 2) simple to be implemented in view of real-time capability and energy conservation, and 3) distributed to avoid being based on global information. After considering these conditions, we propose an Action Force Method, whose details are presented in the following.

6.2. Action Force Method

The Action Force Method is inspired by the theory of universal gravitation in physics [Verlinde 2011]. Suppose that there are interactive forces among mobile nodes and between mobile nodes and targets. For a single mobile node i, we consider these interactive forces to it in its adjacent area within range $2d_0$. Assume that target k exerts its attractive force from itself to mobile node i, which is expressed as

$$F'_{ki} = G \times \frac{1}{d_{ki}^2} \tag{8}$$

where G stands for the gravitational acceleration which is a constant and d_{ki} represents the Euclidean distance between target k and mobile node i.

Assume that there are other N mobile nodes in the adjacent area of mobile node i. Among them, mobile node j ($j \neq i$) exerts its repulsive force to node i, which is expressed as

$$F_{ji} = -G \times \frac{\cos\theta_j}{d_{ji}^2} \tag{9}$$

where d_{ji} represents the Euclidean distance between node j and node i; θ_j stands for the included angle between the line from node i to node j and the line from node i to target k. So the direction of the repulsive force is along the way from target k to node i, which is opposite to the attractive force. Therefore, the negative sign before the formula is needed. For all N mobile nodes, the total repulsive force is

$$F_{Ni} = \sum_{j=1, j\neq i}^{N} F_{ji} = -G \times \sum_{j=1, j\neq i}^{N} \frac{\cos\theta_j}{d_{ji}^2} \ . \tag{10}$$

Therefore, the action force from target k to mobile node i is defined as

$$F_{ki} = \eta F'_{ki} + (1-\eta) F_{Ni} \tag{11}$$

where η is a adjustment factor. The factor η is a percentage number from 0 to 1 (100%). It reflects how much percentage of attractive force from the target represents in the action force. In order to make mobile nodes evenly distributed among all targets, the factor η could be a small number such as 0.1 or 0.2. In this case, if there has been several mobile nodes around target k, node i will be likely to choose other target for tracking because the attractive force from target k to node i is much weakened by the repulsive force from other nodes. In this way, it can be effectively prevented that there are a few of targets not tracked by any mobile node.

Assume that there are K candidate targets for mobile node i to select, the actual target T for mobile node i to track is

$$T = \arg\max_{k\in K} F_{ki} \ . \tag{12}$$

Algorithm 2 Multi-Mobile Tracking based on Tacit Agreement (*MTTA-M*)

repeat
  for fixed node i **do**
    Chooses the nearest sensed target to detect;
  end for
  for mobile node m **do**
    Node m senses K targets within its $2d_0$ range;
    for those sensed target $k \in K$ **do**
      Node m computes the attractive force F'_{km} according to Formula (8);
      for other mobile node $j \in N$ within range $2d_0$ of m **do**
        Node m gets F_{jm} according to Formula (9);
        $F_{Nm} = F_{Nm} + F_{jm}$;
      end for
      Node m computes the action force F_{km} from k according to Formula (11);
    end for
    Node m chooses the appropriate target T based on Formula (12);
  end for
  Execute MTTA to track the targets which are seleted;
until no sensed targets

6.3. Multiple Targets Tracking Algorithm

The multiple targets tracking algorithm is called *MTTA-M* (Multi-MTTA), which is based on MTTA and advanced with action force mothed. When Θ targets appear in the deployment area, the entire network implements this solution to track them. In every moment, one node can track only one target, so a fixed node just detects the nearest target in its vicinity. For a single target k, some nodes in the active state try to detect it and keep the tracking probability P_{Dk} not less than the threshold α. More importantly, mobile nodes follow these targets for tracking. In a work cycle, a mobile node chooses one target for tracking based on the action force mothed, which means it tracks the target that exerts the maximum action force to it. As mobile nodes move closer to targets, some fixed nodes can switch to the sniff state for energy saving , meanwhile the tracking probability for targets can be guaranteed. In contrast, if P_{Dk}

of any target k is not enough, more fixed nodes will switch to the active state and track the target. The whole process can be seen in Algorithm 2.

7. PERFORMANCE EVALUATION

In this section, extensive simulations results are presented in order to verify the performance of our proposed solutions. We conduct our simulations in NS-2. In the simulation scenario, nodes are randomly deployed in a surveillance area which is 100×100 m^2. There are two parts for all simulations. The first part is for single target tracking solutions, and the second is for multiple targets tracking solutions, where there are 10 targets in the area. Table II shows some parameters for the basic network setup. Targets in simulations appear in arbitrary position of the area, and move in a random direction for some time and then pause for a while, as mentioned in Section 3. Figure 9 is an illustration of one simulation scenario. It is a snapshot from the software Nam, which is an animation tool for viewing network simulation traces. The target, mobile nodes and fixed nodes are represented by hexagon, square and circle, respectively.

Table II: Simulation Parameters

Parameters	Values
area size (m^2)	100×100
single simulation time (s)	100
number of one experiment	100
s—the number of fixed nodes	100
n—the number of mobile nodes	10
v_m—the velocity of mobile nodes (m/s)	1
the minimal velocity of target(s) (m/s)	0.7
v_t—the maximal velocity of target(s) (m/s)	3
μ—the mean value of noise (dB)	0.1
σ—the standard deviation of noise (dB)	0.2
S_0—measurement within curtain distance (dB)	5
k—the decaying factor	2
d_0—distance threshold for measurement (m)	6
λ—measurement threshold for detection (dB)	5
α—the threshold of tracking probability (%)	80%
η—the adjustment factor for action force	0.1

For comparison, there are four different kinds of solutions for target tracking being conducted. The first is our proposed solution MTTA. The second is the Fixed Tracking (FT), which is a kind of traditional solution based on fixed nodes such as [Khedr and Osamy 2011]. The third kind of method is the Cluster Tracking (CT), which assumes that those fixed nodes around the target can form a cluster such as in [Wang et al. 2013]. The nodes in the cluster keep active to track the target. The methods FT and CT are two typical kinds of methods based on fixed nodes. The last one is the Hole-Filling Tracking (HFT), which focuses on deploying mobile nodes to fix coverage holes. This kind of solution deploys mobile nodes to move into coverage holes in order to fix them such as in [Lin and Tang 2011]. The method HFT is a typical kind of tracking methods based on hybrid networks.

The metrics we use in simulation experiments are *the effective monitoring ratio* and *the energy consumption*. Note that the effective monitoring time is the total time when the tracking probability is not less than the threshold α during the simulation. Therefore, the effective monitoring ratio is the quotient of total effective monitoring time divided by the simulation time. A higher effective monitoring ratio reflects a higher degree of effectiveness or higher quality for tracking solutions. On the other hand, energy consumption is in common use for solution evaluation in WSNs. As mobile nodes

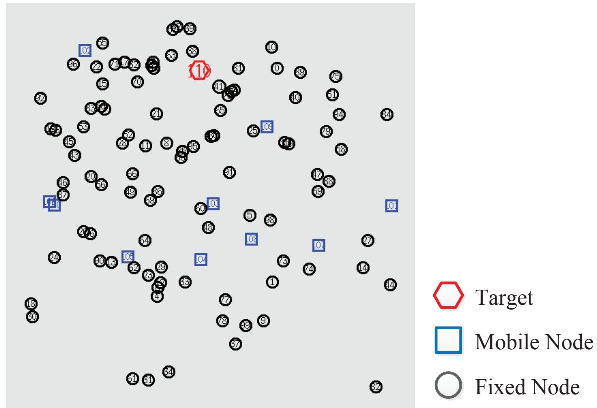


Fig. 9: A snapshot of the simulation scenario.

are less constrained in energy and we focus on reducing the energy consumed by fixed nodes, the energy consumption is equal to the sum of active fixed nodes. We assume that all fixed nodes can switch between active and sniff states, namely, the nodes are in the active state only when they track targets, and the power of fixed nodes in the active state and the sniff state is 1 mJ/s and 0 mJ/s, respectively.

7.1. Single Target Tracking Simulations

7.1.1. Number of Mobile Nodes. Figure 10 shows the effective monitoring ratios achieved by MTTA, FT, CT and HFT when the number of mobile nodes increases from 0 to 20. When the number is 0, CT has the best performance among these four solutions, because a cluster is more likely to cover all possible directions of the target. However, their ratio can hardly reach 50%. As shown in the figure, with the increase in mobile nodes, the performance of MTTA and HFT appear to rise, and MTTA achieves much better. To be specific, when the number of mobile nodes is up to 20, MTTA's ratio is about 30% greater than HFT's and about 50% greater than FT's and CT's, respectively. It shows that the participation of mobile nodes significantly improves the tracking effectiveness.

Figure 11 shows the energy consumption of MTTA, FT, CT and HFT. For MTTA and HFT, as mobile nodes increase, the energy consumed by fixed nodes in the entire network is reduced, which is much more obvious for MTTA. When the number of mobile node is 20, the energy consumption of MTTA is half of other solutions' at most. As mobile nodes can move close to the target and detect it at the same time, the tracking work needed to be done by a large number of fixed nodes can be finished just by several mobile nodes. Therefore, a certain number of fixed nodes can be released to the sniff state and the energy consumption of them can be reduced sharply.

7.1.2. Number of Fixed Nodes. Figure 12 shows the changes in effective monitoring ratios of MTTA, FT, CT and HFT when the density of fixed nodes changes. With the increase in the number of fixed nodes, the performances achieved by all solutions gradually trend upward, because the increased density of fixed nodes provides better coverage in the field for target detection. When the number of fixed nodes is 40, the effective monitoring ratio of FT or CT is less than 20%, and HFT doesn't work well, neither, because holes seems to be everywhere in such situation. It reflects that, in fixed sensor networks, the tracking ability is mainly depended on deployment density of sensors. The participation of mobile nodes can overcome this defect to some extent.

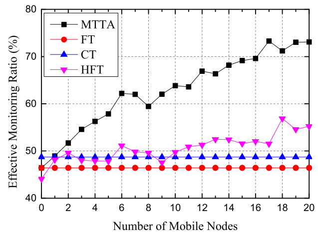


Fig. 10: Number of mobile nodes vs. effective monitoring ratio.

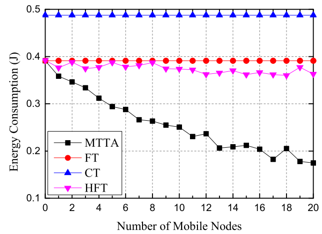


Fig. 11: Number of mobile nodes vs. energy consumption.

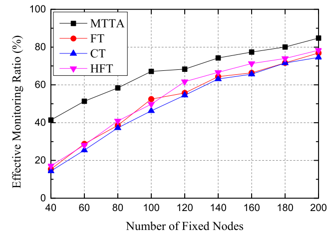


Fig. 12: Number of fixed nodes vs. effective monitoring ratio.

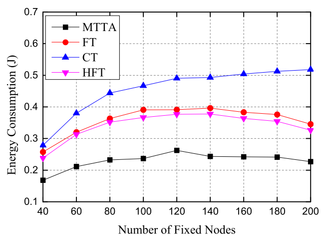


Fig. 13: Number of fixed nodes vs. energy consumption.

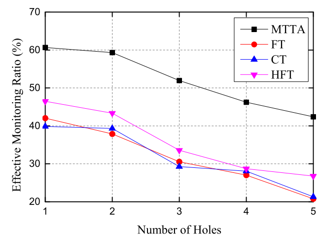


Fig. 14: Number of holes vs. effective monitoring ratio.

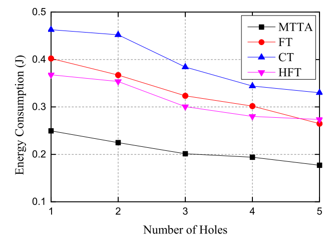


Fig. 15: Number of holes vs. energy consumption.

Figure 13 shows the energy consumption of MTTA, FT, CT and HFT. With the increase in the number of fixed nodes, energy consumed by all solutions are increasing in the first half. But for the last half, the increase of CT's almost stops, and others even begin to decrease. The reason is that, in the beginning, the fixed nodes are rare and the nodes involved in monitoring are few in number, leading to the low power consumption. This also causes a low effective monitoring ratio. After that, with the increase in the number of fixed nodes, more nodes can be involved in monitoring, hence the energy consumption begin to increase. However, when the number of fixed nodes increases to some degree, the energy consumption begins to decrease, because there are more opportunities to choose closer nodes to the target to guarantee the detection probability. Therefore, the number of nodes involved in tracking becomes smaller and then the energy consumption is in decline.

7.1.3. Number of Holes. Figure 14 shows the effective monitoring ratios of four solutions when the number of coverage holes increases from 1 to 5. Obviously, with the increase in the number of holes, performances achieved by four solutions all trend downward, because there are no fixed nodes deployed in holes, once the target moves into one of them, it is difficult to be detected by nodes. When the number of holes increases to 5, the ratio achieved by MTTA is twice as high as that of FT or CT and 50% more than that of HFT. This result shows that MTTA is less prone to miss the target, as mobile nodes can follow it even in holes and alleviate the adverse effect of holes to some extent.

Figure 15 shows energy consumption of four solutions is influenced by the number of coverage holes. It can be seen that results of four solutions all decrease. This figure looks somewhat like the effective monitoring ratio one. However, the difference is that the sequence of result curves is almost reversed, as MTTA's energy consumption is the lowest among all solutions. The reason why the energy consumption decrease is that

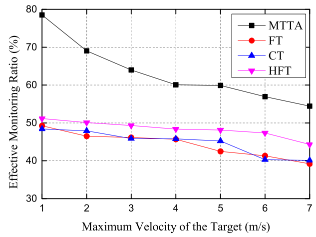


Fig. 16: Maximum velocity of the target vs. effective monitoring ratio.

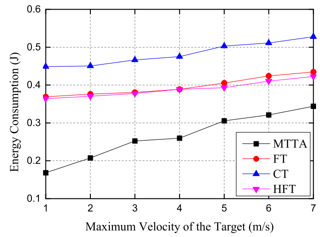


Fig. 17: Maximum velocity of the target vs. energy consumption.

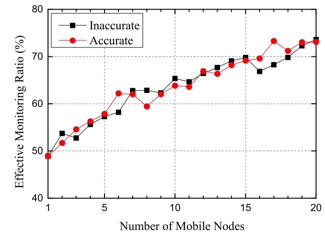


Fig. 18: Number of mobile nodes vs. effective monitoring ratio.

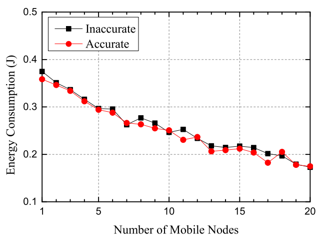


Fig. 19: Number of mobile nodes vs. energy consumption.

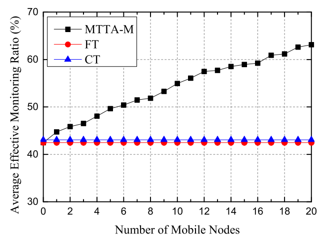


Fig. 20: Number of mobile nodes vs. average effective monitoring ratio.

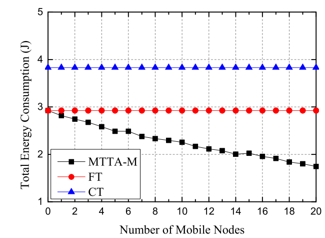


Fig. 21: Number of mobile nodes vs. total energy consumption.

when the target is in a hole where few nodes can detect it, the number of active nodes becomes smaller. So the energy consumption for the whole network decreases.

7.1.4. Maximum Velocity of the Target. Figure 16 shows effective monitoring ratios of four solutions are influenced by the change of maximum velocity of the target. When the maximum velocity increases, performances of four solutions decrease more or less. Among them, MTTA is affected most seriously, because it is difficult for mobile nodes to follow a fast moving target. Even so, the result shows that MTTA is still effective when the maximum velocity of the target is up to 7 m/s, as its ratio outperforms FT or CT by 30% and HFT by 20%.

Figure 17 shows how the energy consumption of four solutions are affected by the maximum velocity of the target. It can be seen that as maximum velocity increases, energy consumption of these solutions increase as well, because when the target moves fast in the area, more nodes need to transit their states frequently, which can cause more energy consumption.

7.1.5. Effects of Inaccurate Target's Positions. We also do the experiment on MTTA with inaccurate location of the target. In this experiment, nodes have the target's position with a 2-meter offset. Figure 18 and Figure 19 show how MTTA is affected by the inaccuracy in the target's position when the number of mobile nodes increases from 1 to 20. In Figure 18, the ratio with accurate location only outperforms that with inaccurate location by a small extent in general. Similarly, in Figure 19, the energy consumption with inaccurate location only generally increases a little. The results show that the performance of MTTA is robust even if there are some offsets in the target's position.

7.2. Multiple Targets Tracking Simulations

7.2.1. Number of Mobile Nodes. Figure 20 shows how average effective monitoring ratios of MTTA-M, FT and CT when the number of mobile nodes increases from 0 to 20. For

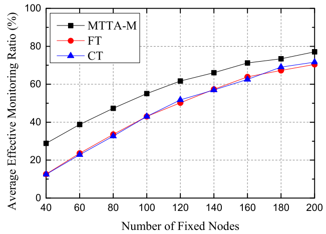


Fig. 22: Number of fixed nodes vs. average effective monitoring ratio.

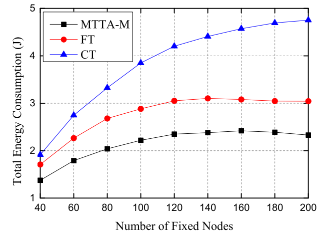


Fig. 23: Number of fixed nodes vs. total energy consumption.

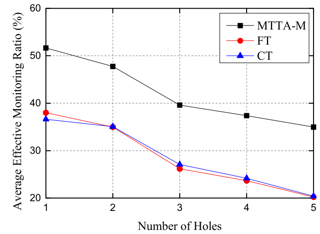


Fig. 24: Number of holes vs. average effective monitoring ratio.

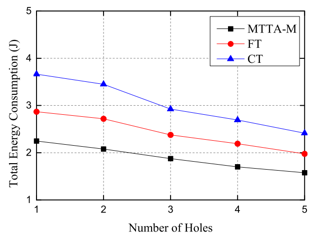


Fig. 25: Number of holes vs. total energy consumption.

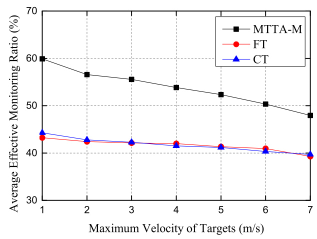


Fig. 26: Maximum velocity of targets vs. average effective monitoring ratio.

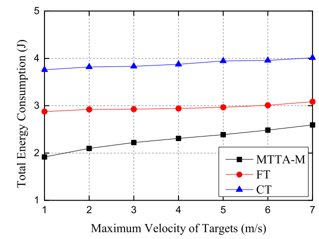


Fig. 27: Maximum velocity of targets vs. total energy consumption.

these solutions, the average ratio is the average of all effective monitoring ratios for all targets. It can be seen that with the increase of mobile nodes, the performance of MTTA-M is on the rise. When the number is 20, the average ratio of MTTA-M is 50% more than that of FT and CT, respectively.

The influence for all solutions upon total energy consumption caused by increasing number of mobile nodes is shown in Figure 21. Total energy consumption is the sum of energy consumed to track all targets. As shown in the figure, the consumption of MTTA-M decreases gradually when the number of mobile nodes increases from 0 to 20. On the other hand, CT consumes much more energy as it needs all nodes in the cluster to be active.

7.2.2. Number of Fixed Nodes. Figure 22 shows how the average effective monitoring ratio is achieved by these three solutions when the number of fixed nodes increases from 40 to 200. The performances achieved by three solutions all trend upward gradually as the number of fixed nodes increases. It can be seen that MTTA-M outperforms FT and CT all the way. It shows that work used to be completed by a large number of fixed nodes can be accomplished by a few mobile nodes.

Figure 23 represents total energy consumption results of MTTA-M, FT and CT. When the number of fixed nodes increases from 40 to 200, energy consumption of all solutions gradually increase. When the number increases to some degree, the results of MTTA-M and FT appear to slide down a little. This phenomenon is similar to the single target tracking scenario. It shows that as the multiple targets situation needs more nodes to achieve curtain tracking quality, the gradient of these curves are slighter. When there are more opportunities to choose closer nodes to targets, total energy consumption begins to decrease.

7.2.3. Number of Holes. Figure 24 shows changes in average effective monitoring ratio of MTTA-M, FT and CT when the number of Holes increases from 1 to 5. Due to the

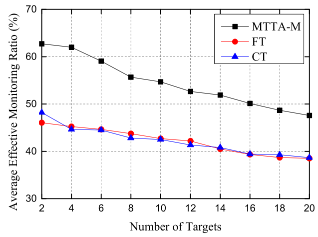


Fig. 28: Number of targets vs. average effective monitoring ratio.

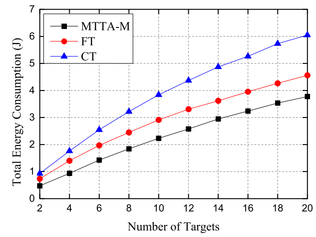


Fig. 29: Number of targets vs. total energy consumption.

increase, the performances achieved by three solutions all trend downward. As we can see, MTTA-M gets a better ratio than others. To be specific, when there are 5 holes, MTTA-M outperforms both two other solutions by more than 70%, respectively.

The same effect for all solutions on total energy consumption is shown in Figure 25. When the number of coverage holes increases from 1 to 5, the total energy consumptions of three solutions decrease gradually. The reason is similar to that of the single target tracking simulation, that is, the number of active nodes becomes smaller when targets are in holes. The difference is that because there are 10 targets in the scenario, the total energy consumption of the multiple targets simulation becomes almost 10 times as much as that of the single target scenario.

7.2.4. Maximum Velocity of Targets. Figure 26 shows how changes in maximum velocity of targets influences the average effective monitoring ratio achieved by MTTA-M, FT and CT. When the maximum velocity increases from 1 m/s to 7 m/s, results of three solutions decrease more or less. It can be seen that the increase in velocity bring negative effect upon both mobile and fixed methods. When maximum velocity of targets is 7 m/s, MTTA-M outperforms FT and CT by about 20%, respectively.

The negative effect from increasing maximum velocity of targets to total energy consumption is also reflected by Figure 27. In this figure, results of three solutions all trend upward when the maximum velocity of targets increases from 1 m/s to 7 m/s. This phenomenon is similar to the single target one. When the maximum velocity is 7 m/s, total energy consumption of MTTA-M is 20% less than that of FT and 37% less than that of CT.

7.2.5. Number of Targets. We also conduct simulations with changes in the number of targets for multiple targets tracking solutions. Figure 28 shows how these three solutions perform when the number of targets increase from 2 to 20. It can be seen that all solutions' results slide down. The reason is that every individual node can only track one target at single time point. Therefore, it is hard to track several targets with limited nodes. In this situation, MTTA-M also achieves best performance among these three solutions. When there are 20 targets, the average ratio of MTTA-M is 25% more than that of FT and CT, respectively.

Results of total energy consumption of all solutions influenced by the number of targets are shown in Figure 29. We can see that when the number of targets increases, these curves in figure trend upward, which means all solutions need to consume more energy, because as more targets appear in the area, more nodes keep in the active state to track them. In this situation, MTTA-M still presents the best performance among these three solutions. Total energy consumption of MTTA-M is 13% and 30% less than that of FT and CT, respectively.

8. CONCLUSION

In this paper, we discuss our improvement for target tracking solutions by using mobile sensors. We introduce a few mobile sensors into traditional WSNs to build up a hybrid sensor network for continuous target tracking. Based on these mobile sensors, we propose a single target tracking solution. By scheduling mobile sensors to follow the target and cooperate with fixed sensors, the tracking probability can be effectively guaranteed as mobile sensors can detect the target in a short distance. The movement path of mobile sensors in this solution is based on provable performance bound compared with the optimal solution. Since a few of mobile sensors can complete the work used to be done by a lot of fixed sensors, the energy consumed by fixed sensors can be effectively saved, and the lifetime of networks is prolonged. Even if there are coverage holes in the surveillance area, mobile sensors can also effectively alleviate the adverse effect by following the target into these hole areas. Moreover, we analyze the multiple targets tracking scenario and extend our solution to fit multiple targets situation. We propose the action force method, in which mobile sensors can choose appropriate targets to track and try not to ignore any one of them. Based on this efficient and distributed method, we propose a multiple targets tracking solution. We verify the effectiveness of our solutions using extensive simulations. Results show that our proposed algorithms can achieve higher effective monitoring ratios and lower energy consumptions.

In our future work, we attempt to consider the energy consumption of mobile nodes during tracking. First, we plan to add some charging stations for mobile nodes in the scenario. Second, we need to revise the problem which considers energy consumption of mobile nodes. Third, we should improve our algorithms to contain the charging process of mobile nodes. Energy efficiency would also be an interesting topic which considers reducing the amount of energy in the movement.

Acknowledgment

Above work was supported in part by grants from the National Natural Science Foundation (NSF) of China under grant Nos. 61572206 and 61305085, the Natural Science Foundation of Fujian Province of China under grant No. 2014J01240, and the Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University (No.ZQN-PY308).

REFERENCES

Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. 2002. Wireless sensor networks: a survey. *Computer Networks* 38, 4 (2002), 393–422.

Edoardo Amaldi, Antonio Capone, Matteo Cesana, and Ilario Filippini. 2012. Design of wireless sensor networks for mobile target detection. *IEEE/ACM Transactions on Networking* 20, 3 (2012), 784–797.

George K Atia, Venugopal V Veeravalli, and Jason A Fuemmeler. 2011. Sensor scheduling for energy-efficient target tracking in sensor networks. *IEEE Transactions on Signal Processing* 59, 10 (2011), 4923–4937.

Jing Bai, Peng Cheng, Jiming Chen, Adrien Guenard, and Yeqiong Song. 2012. Target Tracking with Limited Sensing Range in Autonomous Mobile Sensor Networks. In *IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 329–334.

MZA Bhuiyan and others. 2010. Prediction-based energy-efficient target tracking protocol in wireless sensor networks. *Journal of Central South University of Technology* 17, 2 (2010), 340–348.

Joe C Chen, Ralph E Hudson, and Kung Yao. 2002. Maximum-likelihood source localization and unknown sensor location estimation for wideband signals in the near-field. *IEEE Transactions on Signal Processing* 50, 8 (2002), 1843–1854.

Wei-Peng Chen, Jennifer C Hou, and Lui Sha. 2004. Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing* 3, 3 (2004), 258–271.

Hung-Chi Chu and Rong-Hong Jan. 2005. A GPS-less self-positioning method for sensor networks. In *11th International Conference on Parallel and Distributed Systems*, Vol. 2. IEEE, 629–633.

Oualid Demigha, W-K Hidouci, and Toufik Ahmed. 2013. On energy efficiency in collaborative target tracking in wireless sensor network: a review. *IEEE Transactions on Control System Technology* 15, 3 (2013), 1210–1222.

Jiangping Hu and Xiaoming Hu. 2010. Nonlinear filtering in target tracking using cooperative mobile sensors. *Automatica* 46, 12 (2010), 2041–2046.

Shiow-Fen Hwang, Kun-Hsien Lu, Liang-Ren Yang, and Chyi-Ren Dow. 2008. Efficient data reporting for object tracking in wireless sensor networks with mobile sinks. In *14th Asia-Pacific Conference on Communications (APCC)*. IEEE, 1–5.

Natallia Katenka, Elizaveta Levina, and George Michailidis. 2008. Local vote decision fusion for target detection in wireless sensor networks. *IEEE Transactions on Signal Processing* 56, 1 (2008), 329–338.

Ahmed M Khedr and Walid Osamy. 2011. Effective target tracking mechanism in a self-organizing wireless sensor network. *J. Parallel and Distrib. Comput.* 71, 10 (2011), 1318–1326.

Ajith S Kumar and Rejina Parvin. 2013. Energy conserving hybrid sensor network for target tracking in Wireless Sensor Networks. In *International Conference on Communications and Signal Processing (ICCSP)*. IEEE, 55–59.

Krzysztof Lembke, L Kietlinski, M Golanski, and Radosław Schoeneich. 2011. RoboMote: Mobile Autonomous Hardware Platform for Wireless Ad-hoc Sensor Networks. In *2011 IEEE International Symposium on Industrial Electronics (ISIE)*. IEEE, 940–944.

Jenn-Wei Lin and Shih-Chieh Tang. 2011. A grid-based coverage approach for target tracking in hybrid sensor networks. *Journal of Systems and Software* 84, 10 (2011), 1746–1756.

Sonia Martínez and Francesco Bullo. 2006. Optimal sensor placement and motion coordination for target tracking. *Automatica* 42, 4 (2006), 661–668.

Farah Mourad, Hicham Chehade, Hichem Snoussi, Farouk Yalaoui, Lionel Amodeo, and Cedric Richard. 2012. Controlled mobility sensor networks for target tracking using ant colony optimization. *IEEE Transactions on Mobile Computing* 11, 8 (2012), 1261–1273.

Marjan Naderan, Mehdi Dehghan, Hossein Pedram, and Vesal Hakami. 2012. Survey of mobile object tracking protocols in wireless sensor networks: a network–centric perspective. *International Journal of Ad Hoc and Ubiquitous Computing* 11, 1 (2012), 34–63.

Ruixin Niu, Pramod K Varshney, Michael Moore, and Dale Klamer. 2004. Decision Fusion in a Wireless Sensor Network with a Large Number of Sensors. In *In Fusion*. Citeseer, 21–27.

Alessandro Nordio and C Chiasserini. 2011. Field reconstruction in sensor networks with coverage holes and packet losses. *IEEE Transactions on Signal Processing* 59, 8 (2011), 3943–3953.

Stefan Ruhrup and Ivan Stojmenovic. 2013. Optimizing communication overhead while reducing path length in beaconless georouting with guaranteed delivery for wireless sensor networks. *IEEE Trans. Comput.* 62, 12 (2013), 2440–2453.

Rahul C Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. 2003. Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks* 1, 2 (2003), 215–233.

Xiaoning Shan and Jindong Tan. 2005. Mobile sensor deployment for a dynamic cluster-based target tracking sensor network. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*. IEEE, 1452–1457.

Xiaohong Sheng and Yu-Hen Hu. 2005. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE Transactions on Signal Processing* 53, 1 (2005), 44–53.

Radu Stoleru, John A Stankovic, and Sang H Son. 2007. Robust node localization for wireless sensor networks. In *Proceedings of the 4th workshop on Embedded networked sensors*. ACM, 48–52.

Rui Tan, Guoliang Xing, Jianping Wang, and Hing Cheung So. 2010. Exploiting reactive mobility for collaborative target detection in wireless sensor networks. *IEEE Transactions on Mobile Computing* 9, 3 (2010), 317–332.

Jing Teng, Hichem Snoussi, and Cédric Richard. 2012. Prediction-based cluster management for target tracking in wireless sensor networks. *Wireless Communications and Mobile Computing* 12, 9 (2012), 797–812.

Hua-Wen Tsai, Chih-Ping Chu, and Tzung-Shi Chen. 2007. Mobile object tracking in wireless sensor networks. *Computer Communications* 30, 8 (2007), 1811–1825.

Gurkan Tuna, V Cagri Gungor, and Kayhan Gulez. 2014. An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters. *Ad Hoc Networks* 13 (2014), 54–68.

Pramod K Varshney. 1996. *Distributed detection and data fusion*. Springer-Verlag New York, Inc.

Erik Verlinde. 2011. On the origin of gravity and the laws of Newton. *Journal of High Energy Physics* 2011, 4 (2011), 1–27.

Pascal Vicaire, Tian He, Qing Cao, Ting Yan, Gang Zhou, Lin Gu, Liqian Luo, Radu Stoleru, John A Stankovic, and Tarek F Abdelzaher. 2009. Achieving long-term surveillance in vigilnet. *ACM Transactions on Sensor Networks (TOSN)* 5, 1 (2009), 9.

Zhibo Wang, Wei Lou, Zhi Wang, Junchao Ma, and Honglong Chen. 2013. A hybrid cluster-based target tracking protocol for wireless sensor networks. *International Journal of Distributed Sensor Networks* 2013 (2013), 1–16.

Thakshila Wimalajeewa and Sudharman K Jayaweera. 2010. Mobility assisted distributed tracking in hybrid sensor networks. In *IEEE International Conference on Communications (ICC)*. IEEE, 1–5.

Guoliang Xing, Minming Li, Tian Wang, Weijia Jia, and Jun Huang. 2012. Efficient Rendezvous Algorithms for Mobility-enabled Wireless Sensor Networks. *IEEE Transactions on Mobile Computing* 11, 1 (2012), 47–60.

Zaiyue Yang, Xiufang Shi, and Jiming Chen. 2014. Optimal Coordination of Mobile Sensors for Target Tracking under Additive and Multiplicative Noises. *IEEE Transactions on Industrial Electronics* 61, 7 (2014), 3459–2468.

Degan Zhang, Guang Li, Ke Zheng, Xuechao Ming, and Zhao-Hua Pan. 2014. An energy-balanced routing method based on forward-aware factor for wireless sensor networks. *IEEE Transactions on Industrial Informatics* 10, 1 (2014), 766–773.

Ke Zhou and Stergios I Roumeliotis. 2008. Optimal motion strategies for range-only constrained multisensor target tracking. *IEEE Transactions on Robotics* 24, 5 (2008), 1168–1185.

Yi Zou and Krishnendu Chakrabarty. 2007. Distributed mobility management for target tracking in mobile sensor networks. *IEEE Transactions on Mobile Computing* 6, 8 (2007), 872–887.