# Collaborative Target Detection in Wireless Sensor Networks with Reactive Mobility

Rui Tan*, Guoliang Xing*, Jianping Wang* and Hing Cheung So†
*Department of Computer Science, City University of Hong Kong
r.tan@student.cityu.edu.hk,{glxing, jianwang}@cityu.edu.hk
†Department of Electronic Engineering, City University of Hong Kong
hcso@ee.cityu.edu.hk

***Abstract*—Recent years have witnessed the deployments of wireless sensor networks in a class of mission-critical applications such as object detection and tracking. These applications often impose stringent QoS requirements including high detection probability, low false alarm rate and bounded detection delay. Although a dense all-static network may initially meet these QoS requirements, it does not adapt to unpredictable dynamics in network conditions (e.g., coverage holes caused by death of nodes) or physical environments (e.g., changed spatial distribution of events). This paper exploits reactive mobility to improve the target detection performance of wireless sensor networks. In our approach, mobile sensors collaborate with static sensors and move reactively to achieve the required detection performance. Specifically, mobile sensors initially remain stationary and are directed to move toward a possible target only when a detection consensus is reached by a group of sensors. The accuracy of final detection result is then improved as the measurements of mobile sensors have higher signal-to-noise ratios after the movement. We develop a sensor movement scheduling algorithm that achieves *near-optimal* system detection performance within a given detection delay bound. The effectiveness of our approach is validated by extensive simulations using the real data traces collected by 23 sensor nodes.**

## I. Introduction

In recent years wireless sensor networks (WSNs) have been deployed in a class of mission-critical applications such as target detection [1], object tracking [2], and security surveillance [3]. A fundamental challenge for these WSNs is to meet stringent QoS requirements including high target detection probability, low false alarm rate and bounded detection delay. However, physical phenomena (e.g., the appearance of intruders) often have unpredictable spatiotemporal distributions. As a result, a large network deployment may require excessive sensor nodes in order to achieve satisfactory sensing performance. Moreover, although dense node deployment may initially achieve the required performance, it does not adapt to dynamic changes of network conditions or physical environments. For instance, death of nodes due to battery depletion or physical attacks can easily cause coverage holes in a monitored battlefield.

In this paper, we exploit reactive mobility to improve the target detection performance of WSNs. In our approach, sparsely deployed mobile sensors collaborate with static sensors and move in a reactive manner to achieve required detection performance. Specifically, mobile sensors remain stationary until a possible target is detected. The accuracy of the final detection decision will be improved after mobile sensors move toward the possible target position and achieve higher Signal-to-Noise Ratios (SNRs). By taking advantage of such reactive mobility, a network can adapt to irregular and unpredictable spatiotemporal distribution of targets. Moreover, the sensor density required in a network deployment is significantly reduced because the sensing coverage can be reconfigured in an on-demand fashion.

Several challenges must be addressed for utilizing the mobility of sensors in target detection. First, practical mobile sensors are only capable of slow-speed movement, which may lead to long detection delays. The typical speed of mobile sensor systems (e.g., NIMs [4], Packbot [5] and Robomote [6]) is about $0.2-2$ m/s. Therefore, the movement of sensors must be efficiently scheduled in order to reduce detection latency. Second, the number of mobile sensors available in a network deployment is often much smaller than that of static sensors due to the higher manufacturing cost. Hence mobile sensors must effectively collaborate with static sensors to achieve the maximum utility. At the same time, the coordination among sensors should not introduce high overhead or significant detection delay. Third, the distance that mobile sensors move in a detection process should be minimized. Due to the high power consumption of locomotion, frequent movement will quickly deplete the battery of a mobile node. For instance, a Robomote [6] sensor needs to recharge every 20 minutes when constantly moving. Although mobile sensors may recharge their batteries by moving to locations with wired power sources, frequent battery recharging causes disruptions to network topologies. Finally, moving sensors lower the stealthiness of a network, which is not desirable for many applications deployed in hostile environments like battlefields.

We attempt to address the aforementioned challenges and demonstrate the advantages of reactive mobility in target-detection WSNs. This paper makes the following major contributions:

- We propose a novel two-phase detection approach that utilizes mobility of sensors to improve detection performance. Mobile sensors initially remain stationary and are directed to move toward a possible target only when a detection consensus is reached by all nearby mobile and static sensors. Such a two-phase strategy allows mobile sensors to avoid unnecessary movement through the coordination with static sensors.

- We develop a *near-optimal* movement scheduling algorithm based on dynamic programming that minimizes the expected moving distance of mobile sensors under a given detection delay bound. Meanwhile, our scheduling algorithm enables mobile sensors to locally control their movement and sensing, thus both coordination overhead and detection delay are reduced significantly.
- We conduct extensive simulations using real data traces collected by 23 sensors in the SensIT vehicle detection and tracking experiments [7]. Our results provide several important insights into the design of target-detection systems with mobile sensors. First, we show that a small number of mobile sensors can significantly boost the detection performance of a network. Second, tight detection delays can be achieved by efficiently scheduling slow-moving mobile sensors.

The rest of the paper is organized as follows. Section II reviews related work. Section III and IV introduce the background and the formulation of our problem. The performance of the proposed two-phase detection model is studied in Section V. Section VI presents a near-optimal movement scheduling algorithm. We present simulation results in Section VII and conclude the paper in Section VIII.

## II. Related Work

Recent work demonstrated that the sensing performance of WSNs can be improved by integrating mobility. Several projects proposed to eliminate coverage holes in a sensing field by relocating mobile sensors [8], [9], [10]. Although such an approach improves the sensing coverage of the initial network deployment, it does not dynamically improve the network's performance after targets of interest appear. Complementary to these projects, we focus on online sensor collaboration and movement scheduling strategies that are used after the appearance of targets.

Several recent studies [11], [12] analyzed the impact of mobility on detection delay and area coverage. These studies are based on random mobility model and do not address the issue of actively controlling the movement of sensors. Bisnik et al. [13] analyzed the performance of detecting stochastic events using mobile sensors. Chin et al. [14] proposed to improve coverage by patrolling fixed routes using mobile sensors. Different from this work, we study efficient sensor collaboration and movement scheduling strategies that achieve specified target detection performance. Reactive mobility is used in a networked robotic sensor architecture [15], [16] to improve the sampling density over a region. However, this project does not focus on target detection under performance constraints.

Collaborative target detection in stationary sensor networks has been extensively studied [17], [1], [18]. The two-phase detection approach proposed in this paper is based on an existing decision fusion model [17]. Several projects studied the network deployment strategies that can achieve specified detection performance [19]. Practical network protocols that facilitate target detection and tracking have also been investigated [3], [20], [21]. Complementary to these studies that deal with the mobility of targets, we focus on improving target detection performance by utilizing the mobility of sensors.

## III. Preliminaries

In this section, we describe the preliminaries of our work, which include the target energy model, the local detection model and the multi-sensor decision fusion model.

### A. Energy Attenuation Model

Sensors detect targets by measuring the energy of signals, e.g., acoustic signal, emitted by targets. The energy attenuates with the distance from the source. Suppose the position of target is at the origin of polar coordinate plane, the attenuated signal energy at the position of sensor $i$ that is $x_i$ meters away from the target is given by

$$e_s(x_i) = \begin{cases} \frac{S_0}{(x_i/d_0)^k} & \text{if } x_i > d_0 \\ S_0 & \text{if } x_i \leq d_0 \end{cases} \tag{1}$$

where $S_0$ is the signal energy measured within $d_0$ from the source, and $d_0$ is a constant determined by target's shape. $k$ is a decaying factor which is typically from 2 to 5. The measurements of sensors are contaminated by background noise which is modeled as zero-mean additive white Gaussian noise with variance of $\varsigma^2$. Accordingly, the energy measurement at sensor $i$ is the sum of signal energy and noise energy $e_n$:

$$e_i = e_s(x_i) + e_n$$

In practice, the energy measurement at a sensor is often estimated by the arithmetic average over a number of samples. If we calculate $e_i$ using $N$ samples, the noise energy $e_n$ is $\frac{1}{N}\sum_{j=1}^{N}\nu^2(j)$, where $\nu(j)$ is the background intensity when taking the $j^{\text{th}}$ sample. $e_n$ follows a Chi-square distribution with mean equal to $\varsigma^2$ and variance equal to $2\varsigma^4/N$. $N$ is often large in practice. As a result, $e_n$ can be approximated by the Gaussian distribution, i.e., $e_n \sim \mathrm{N}(\varsigma^2, 2\varsigma^4/N)$. For example, acoustic data is recorded at a frequency of 4960Hz in the SensIT experiments conducted in [7]. If the energy measurement is calculated every $0.75\,\text{s}$, $N$ is $4960 \times 0.75 = 3720$. We denote $\mu = \varsigma^2$ and $\sigma^2 = 2\varsigma^4/N$ in the remainder of this paper. Consequently, the energy measured at sensor $i$ using $N$ samples follows a Gaussian distribution with the mean decaying with the distance from the source:

$$e_i \sim \mathrm{N}(\mu + e_s(x_i), \sigma^2)$$

### B. Detection and Decision Fusion Models

Data fusion [17] is a widely used technique for improving the performance of detection systems. There exist two basic data fusion schemes, namely, value fusion and decision fusion. In value fusion [22], each sensor sends its raw energy measurements to the cluster head, which makes the detection decision based on the received energy measurements. Different from value fusion, decision fusion operates in a distributed manner as follows. Each sensor makes a *local* decision based on its measurements and sends its decision to the cluster head, which makes a *system* decision according to the local decisions. Due to its low overhead, decision fusion is preferred in the bandwidth constrained WSNs. Moreover, decision fusion allows mobile sensors to locally control their movement and sensing, as we will show in Section IV.

Many fusion rules have been proposed in the literature [17] for different detection systems. In this work, we adopt the majority rule due to its simplicity. Specifically, in the local detection, each individual sensor $i$ makes a local decision (0 or 1) by comparing the energy measurement against a detection threshold, $\lambda_i$, and reports its local decision to the cluster head. The cluster head makes the system decision by the majority rule, i.e., if more than half of sensors vote 1, the cluster head decides 1, otherwise, decides 0.

The false alarm rate is the probability of making a positive decision when the target is actually absent, and the detection probability is the probability that a target is correctly detected. We assume a Constant False Alarm Rate (CFAR) detection model [17]. The local detection at sensor $i$ is to test the following hypothesis:

$$H_0 : p(e_i|H_0) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(e_i-\mu)^2}{2\sigma^2}\right)$$
$$H_1 : p(e_i|H_1) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(e_i-\mu-e_s(x_i))^2}{2\sigma^2}\right)$$

where $e_i$ represents the energy measurement using $N$ samples at sensor $i$. $H_0$ and $H_1$ represent the hypothesis that the target is absent and present, respectively. The optimal decision rule is Likelihood Ratio Test [17], in which node $i$ compares its energy measurement $e_i$ with a detection threshold $\lambda_i$. Node $i$ decides 1 if its energy measurement exceeds $\lambda_i$; otherwise, decides 0. So the local false alarm rate $P_F^i$ and the local detection probability $P_D^i$ are given by

$$P_F^i = \int_{\lambda_i}^{+\infty} p(e_i|H_0)\mathrm{d}e_i = Q\left(\frac{\lambda_i - \mu}{\sigma}\right) \quad (2)$$
$$P_D^i = \int_{\lambda_i}^{+\infty} p(e_i|H_1)\mathrm{d}e_i = Q\left(\frac{\lambda_i - \mu - e_s(x_i)}{\sigma}\right) \quad (3)$$

where $Q(\cdot)$ is the complementary cumulative distribution function of standard Gaussian distribution,

$$Q(x) = \int_{x}^{+\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t$$

Obviously, the closer the sensor is from the source, the higher detection probability it will achieve.

Suppose there are total $n$ sensors in a detection cluster, the system false alarm rate $P_F$ and the system detection probability $P_D$ can be expressed as

$$P_F = \Pr(Y \geq \frac{n}{2}|H_0), \quad P_D = \Pr(Y \geq \frac{n}{2}|H_1)$$

where $Y$ represents the total number of positive local decisions. Denote $I_i$ as the local decision of node $i$, then $Y = \sum_{i=1}^{n} I_i$.

## IV. Mobility-assisted Target Detection with Decision Fusion

This section formulates our problem called the Mobility-assisted Detection with Decision Fusion. The network model is described in Section IV-A. In Section IV-B, a two-phase detection approach is proposed. A numerical example is given in Section IV-C. The problem is formally formulated in Section IV-D.

### A. Network Model

The network is composed of a number of static and mobile sensors. Targets appear at a set of known physical locations referred to as *surveillance spots* with certain probabilities. Surveillance spots are often identified by the network autonomously after the deployment. Therefore, it is impossible to deploy sensors only around surveillance spots. Nodes in the network self-organize into clusters around the surveillance spots by running a clustering protocol [21], such that each cluster monitors a surveillance spot. The above surveillance model is consistent with several previous works [23], [24]. We assume that each static sensor belongs to only one cluster. However, a mobile sensor may belong to multiple clusters because it can contribute to the detection at different surveillance spots.

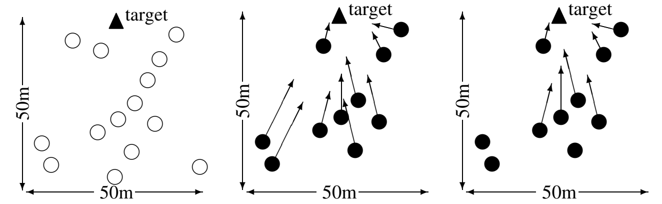
We now briefly discuss how the above network model can be applied to a target detection application. Suppose a number of mobile and static sensors are randomly deployed (e.g., dropped off from an aircraft) in a battlefield to detect military targets. After operating for a certain amount of time, the network may identify some important locations (e.g., based on detection history) as surveillance spots. A cluster is then formed around each spot to perform the detection.

### B. A Two-phase Detection Model

As each cluster performs detection separately, our discussion focuses one cluster hereafter. We design a two-phase detection approach to utilize the mobility of sensors.

In the first phase, all sensors synchronously measure energy and make local decisions by comparing against a predefined threshold. Each sensor reports its local decision to the cluster head, which makes a system decision according to the majority rule. If a positive system decision is made, the second-phase detection is initiated.

In the second phase, each mobile sensor moves toward the surveillance spot according to its movement schedule composed of a list of *moves*. A move specifies the distance a mobile sensor should move and the time instance it starts to move. Both fixed sensors and mobile sensors measure energy at sampling interval of $T$, and sum all the measured energies up. A sequential fusion-like procedure is adopted. At each sampling interval, if the current sum exceeds a threshold at sensor $i$, a positive local decision is made and reported to the cluster head; otherwise, the sensor continues to sense, and continues to move according to its movement schedule if it is a mobile sensor. Note that a sensor may terminate its detection before the end of the second phase.




(a) 14 static sensors; detection probability is 75%; delay is $2s$ (b) 10 mobile sensors; detection probability is 75%; delay is $7s$ (c) 7 mobile sensors; detection probability is 75%; delay is $15s$

Fig. 1. A numerical example of target detection using static or mobile sensors.

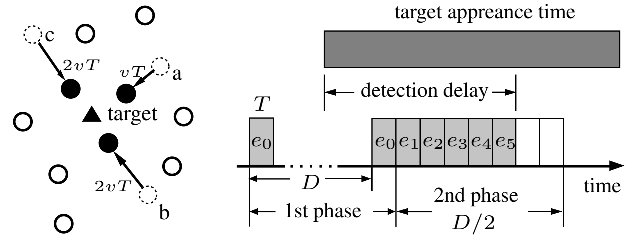
At the end of the second phase, if the energy sum doesn't exceed the threshold, a negative local decision is made and reported. A sensor can terminate its detection before the end of the second phase if it has enough evidence to make a positive local decision. The final system decision is made by the cluster head as soon as enough local decisions are received to reach a majority consensus. After the end of the second phase, the mobile sensors shared by multiple clusters may need to move back to their original positions if such movement break the detection performances of other clusters. Otherwise, these shared mobile sensors stay at the new positions to avoid the energy consumed in moving back.

Such a two-phase approach has several advantages: (1) unnecessary movement of mobile sensors is avoided, as mobile sensors start to move only after the first-phase detection produces a positive decision; (2) the sequential fusion strategy allows each mobile sensor to locally control its sensing and moving according to its movement schedule, which avoids inter-node coordination overhead. Moreover, a mobile sensor may terminate its detection once it has enough evidence to make a positive decision. As a result, the delay of reaching a consensus in the cluster can be reduced.

### *C. A Numerical Example*

We now illustrate our problem and the basic approach using a numerical example. To simplify the discussion, we assume that there is only one surveillance spot. We also assume that, after a possible target appears, a decision consensus is always reached in the first phase of detection, which triggers all mobile sensors to move toward the surveillance spot. The required detection probability and false alarm rate are 75% and 5%, respectively. The minimum movement speed of mobile sensors is 1 m/s. During initialization, the cluster head estimates the parameters of target energy model (see (1)) using a real data set obtained from [7] (the details are given in Section VII).

We now discuss three different cases: a) if all sensors are static, 14 sensors will be needed to achieve the required detection performance within a delay of 2 second as shown in Figure 1(a). b) If the allowable detection delay is 7 seconds, 10 mobile sensors will be needed as they can move closer to the target resulting in higher signal-to-noise ratios. c) If a detection delay of 15 seconds is allowed, only 7 mobile sensors are needed as illustrated in Figure 1(c). This is because these sensors are able to move a longer distance toward the surveillance spot than in case b).




(a) Spatial view: void and solid circles represent fixed and mobile sensors, respectively. The moving distance of a mobile sensor is multiple of $vT$. (b) Temporal view: in the first-phase detection, all sensors sample for $T$ time at a period of $D$; in the second-phase detection, sensors may terminate the detection in advance, e.g., the sensor successfully detects the target and terminates its second phase at the end of the 5th sampling interval, as illustrated.

Fig. 2. The spatial and temporal views of the two-phase detection

Two important observations can be made from this example. First, the number of total sensors can be significantly reduced by taking the advantage of mobility of sensors. Second, scheduling more mobile sensors to move toward a possible target results in a shorter delay. This observation is particularly important as most mobile sensor systems have low movement speeds. Our objective is to find a movement schedule of mobile sensors that minimizes the total moving distance of sensors while achieving the required detection performance and delay.

### *D. Problem Formulation*

Our problem is characterized by a 3-tuple $< \alpha, \beta, D >$. Specifically, for any target appears at the surveillance spot, the objective is to minimize total expected moving distance of mobile sensors subject to the constraints: (1) the system false alarm rate is no higher than $\alpha$; (2) the system detection probability is no lower than $\beta$; and (3) the expected detection delay is no longer than $D$. As discussed in Section I, The objective of minimizing the total moving distance is motivated by several practical considerations including high power consumption of locomotion and disruptions to the network topology caused by sensor movement.

We make the following assumptions in the problem formulation. (1) The surveillance spot is at the origin, the initial position of sensor $i$ is $x_i^0$. (2) The probability that a target appears at the surveillance spot is $P_a$, which is known or can be easily estimated by detection history. And the time that a target remains at the surveillance spot after appearance is much longer than $D$. (3) A mobile sensor moves at a constant speed of $v$, and the moving distance is multiple of $vT$. A sensor move is denoted as $\mathcal{M}_i(x, j)$, which represents mobile sensor $i$'s moving process from position $x$ to $x - vT$ in the $j^{\text{th}}$ sampling interval $[(j-1)T, jT]$. A movement schedule, $\mathbf{S} = \{\mathcal{M}_i(x, j)\}$, is composed of a list of moves.

To ensure a detection delay of $D$, the sampling period in the first-phase detection must be no longer than $D$. To simplify our discussion, we assume that the period equals to

$D$. Hence, the expected delay of the first phase is $\frac{D}{2}$ if the time instance at which the target appears is uniformly distributed within $D$. Accordingly, the duration of the second phase must be no longer than $\frac{D}{2}$ in order to bound the total expected detection delay within $D$. We assume $\frac{D}{2}$ is multiple of $T$. Therefore, there are maximum $\frac{D}{2T}$ sampling intervals in the second phase. Consequently, the constraint of detection delay bound is satisfied. Note that sampling interval $T$ is usually very small in practice. For instance, $T$ is 0.75s in the SensIT experiments [7]. Thus, this assumption has little impact on the temporal precision of detection. The spatial and temporal views of the two-phase detection are illustrated in Figure 2.

We define the following notation. $\lambda_1$ and $\lambda_2$ are the local detection thresholds for the first-phase detection and the second-phase detection, respectively. Note that all sensors use same local detection threshold in each phase, which is derived in Section V. Denote $P_{F_1}$ ($P_{F_2}$) and $P_{D_1}$ ($P_{D_2}$) as the system false alarm rate and the system detection probability in the first (second)-phase detection, respectively. Because a mobile sensor may stop moving before delay bound $D$, the actual number of moves in a detection is a random variable which depends on the detection threshold and movement schedule in the second phase. Let $\mathcal{L}_0(\lambda_2, \mathbf{S})$ and $\mathcal{L}_1(\lambda_2, \mathbf{S})$ represent the total expected number of moves of all mobile sensors when the target is absent and present, respectively. $\mathcal{L}_0(\lambda_2, \mathbf{S})$ and $\mathcal{L}_1(\lambda_2, \mathbf{S})$ are derived in Section V-B.

Our objective is to find a solution, $< \lambda_1, \lambda_2, \mathbf{S} >$, such that the total expected distance that the mobile sensors move away from their original positions is minimized. Formally, the following cost function is minimized:

$$c(\lambda_1, \lambda_2, \mathbf{S}) = (1-P_a) \cdot P_{F_1} \mathcal{L}_0(\lambda_2, \mathbf{S}) + P_a \cdot P_{D_1} \mathcal{L}_1(\lambda_2, \mathbf{S}) \quad (4)$$

subject to the following constraints:

$$P_{F_1} \cdot P_{F_2} \leq \alpha \quad (5)$$
$$P_{D_1} \cdot P_{D_2} \geq \beta \quad (6)$$
$$\lambda_1 \in \Lambda_1 = \{\lambda_{1(1)}, \lambda_{1(2)}, \ldots, \lambda_{1(k)}\} \quad (7)$$
$$\lambda_2 \in \Lambda_2 = \{\lambda_{2(1)}, \lambda_{2(2)}, \ldots, \lambda_{2(k)}\} \quad (8)$$
$$\forall \mathcal{M}_i(x, j) \in \mathbf{S}, (vT \leq x \leq x_i^0) \wedge (1 \leq j \leq \frac{D}{2T}) \quad (9)$$

In the objective function (4), the second phase detection is initiated with the probability of $P_a \cdot P_{D_1}$ if the target is present, and $(1 - P_a) \cdot P_{F_1}$ if the target is absent. (5) and (6) are the detection performance required by user. As the decisions of two phases are mutually independent, the joint false alarm rate and detection probability are product of two phases' false alarm rates and detection probabilities, respectively. (7) and (8) specify discrete values of the two detection thresholds. In practice, the achievable precision of sensors is low-bounded. (9) specifies the spatial and temporal constraints of sensor movement in the second-phase detection. Each mobile sensor must move between its initial position and the surveillance spot, and the movement must complete within $\frac{D}{2}$, which ensures the detection delay bound as discussed above.

## V. Performance Modeling of Two-phase Detection

We now derive the false alarm rates and the detection probabilities in the two phases of detection which are used in Section VI to find the solution of our problem.

### A. First-phase Detection

Since the local false alarm rate doesn't depend on sensor's position, all sensors have same local false alarm rate $\alpha_1 = Q(\frac{\lambda_1 - \mu}{\sigma})$ according to (2). In absence of target, the number of positive local decisions, $Y$, follows a Binomial distribution. Therefore, the system false alarm rate can be calculated as

$$P_{F_1} = \sum_{i=\frac{n}{2}}^{n} \binom{n}{i} \alpha_1^i (1 - \alpha_1)^{n-i}$$

According to de Moivre-Laplace Theorem [25], the Binomial distribution is approximately a Gaussian distribution with mean of $n\alpha_1$ and variance of $n\alpha_1 - n\alpha_1^2$ if $n \geq 10$ [26]. This condition can be met in many moderate to large scale network deployments. Therefore, the system false alarm rate can be approximated by

$$P_{F_1} \simeq Q\left(\frac{\frac{n}{2} - n\alpha_1}{\sqrt{n\alpha_1 - n\alpha_1^2}}\right) \quad (10)$$

We now derive the system detection probability in the first-phase detection. The local decision at sensor $i$, $I_i$, follows a Bernoulli distribution with $\beta_{1,i}$ as the probability of success, where $\beta_{1,i}$ is the local detection probability of sensor $i$ at its original position $x_i^0$ in the first phase. According to (3), $\beta_{1,i} = Q(\frac{\lambda_1 - \mu - e_s(x_i^0)}{\sigma})$. As $I_1, \ldots, I_n | H_1$ are mutually independent, the mean and variance of $Y|H_1$ are as follows,

$$\mathbb{E}[Y|H_1] = \sum_{i=1}^{n} \mathbb{E}[I_i|H_1] = \sum_{i=1}^{n} \beta_{1,i}$$
$$\mathrm{Var}[Y|H_1] = \sum_{i=1}^{n} \mathrm{Var}[I_i|H_1] = \sum_{i=1}^{n} \beta_{1,i} - \sum_{i=1}^{n} \beta_{1,i}^2$$

However, $I_1, \ldots, I_n | H_1$ are not identically distributed, as the local detection probability depends on sensor's position. According to Lyapunov's Central Limit Theorem [27], if the Lyapunov condition is satisfied, the distribution of $Y|H_1$ approaches a Gaussian distribution when $n$ is large. The proof of satisfaction of the Lyapunov condition is omitted here due to space limit and can be found in [28]. Consequently, the system detection probability can be calculated by

$$P_{D_1} \simeq Q\left(\frac{\frac{n}{2} - \sum_{i=1}^{n} \beta_{1,i}}{\sqrt{\sum_{i=1}^{n} \beta_{1,i} - \sum_{i=1}^{n} \beta_{1,i}^2}}\right) \quad (11)$$

### B. Second-phase Detection

In this section, we analyze the performance of the second-phase detection, which includes the false alarm rate, the detection probability, and the expected number of moves under a given movement schedule.

We assume all sensors in the second-phase detection have the same detection threshold of $\lambda_2$. In absence of target, the local false alarm rate is given by

$$\alpha_2 = 1 - \Pr(\bigcap_{j=1}^{\frac{D}{2T}} E_{i,j} < \lambda_2 | H_0) \tag{12}$$

where $E_{i,j} = \sum_{k=1}^{j} e_{i,k}$ and $e_{i,j}$ is the energy received during the $j^{\text{th}}$ sampling interval $[(j-1)T, jT]$ at sensor $i$. As $e_{i,j}$ are independent and identically-distributed (i.i.d.) when target is absent, i.e., $e_{i,j}|H_0 \sim \mathrm{N}(\mu, \sigma^2)$, all sensors share the same local false alarm rate. The joint probability in (12) can be calculated numerically by the Monte-Carlo method. The details are omitted here due to space limit and can be found in [28]. Similar to (10), the system false alarm rate for the second phase can be calculated by

$$P_{F_2} \simeq Q\left(\frac{\frac{n}{2} - n\alpha_2}{\sqrt{n\alpha_2 - n\alpha_2^2}}\right) \tag{13}$$

Similar to (12), the local detection probability of sensor $i$ is

$$\beta_{2,i} = 1 - \Pr(\bigcap_{j=1}^{\frac{D}{2T}} E_{i,j} < \lambda_2 | H_1) \tag{14}$$

For a fixed sensor, the energies received in different sampling intervals are i.i.d.. However, for a mobile sensor, the energies received in different sampling intervals have different mean values, which depend on the movement schedule. Specifically,

$$e_{i,j}|H_1, \text{fixed} \sim \mathrm{N}(\mu + e_s(x_i^0), \sigma^2), \quad \forall j \tag{15}$$

$$e_{i,j}|H_1, \text{mobile} \sim \mathrm{N}(\mu + \xi_{i,j}(\mathbf{S}), \sigma^2), \quad \forall j \tag{16}$$

where $\xi_{i,j}(\mathbf{S})$ is the energy received by mobile sensor $i$ in the $j^{\text{th}}$ sampling interval of the second phase under movement schedule $\mathbf{S}$. Suppose $L_i$ is the number of total moves of sensor $i$ in movement schedule $\mathbf{S}$. As we show in Section VI-A, the system detection performance is likely to be maximized when the $L_i$ moves are consecutive from the beginning of the second-phase detection. In such a case, $\xi_{i,j}(\mathbf{S})$ can be calculated as:

$$\xi_{i,j}(\mathbf{S}) = \begin{cases} \frac{1}{vT}\int_{x_i^0 - jvT}^{x_i^0-(j-1)vT} e_s(x_i)\mathrm{d}x_i & 1 \le j \le L_i \\ e_s(x_i^0 - L_i vT) & j > L_i \end{cases} \tag{17}$$

In (17), when mobile sensor $i$ is moving from $x_i^0 - jvT$ to $x_i^0 - (j-1)vT$ during the $j^{\text{th}}$ sampling interval, the received energy is the arithmetic average over $N$ samples, which can be approximated by the mean value of the function $e_s(x_i)$ defined by (1); when mobile sensor $i$ stops moving and remains at the position of $x_i^0 - L_i vT$ after $L_i$ sampling intervals, the received energy in one sampling interval is a constant, i.e., $e_s(x_i^0 - L_i vT)$. Similar to (11), the system detection probability for the second-phase detection can be calculated by

$$P_{D_2} \simeq Q\left(\frac{\frac{n}{2} - \sum_{i=1}^{n}\beta_{2,i}}{\sqrt{\sum_{i=1}^{n}\beta_{2,i} - \sum_{i=1}^{n}\beta_{2,i}^2}}\right) \tag{18}$$

Denote $l_i$ as the random variable of the actual moves of mobile sensor $i$ in the second phase. We now derive the expected values of $l_i$ when the target is absent and present, which are denoted by $\mathcal{E}_0^i(L_i)$ and $\mathcal{E}_1^i(L_i)$, respectively.

When the target is absent, mobile sensor $i$ terminates the second phase at the end of the $k^{\text{th}}$ sampling interval with probability of

$$\begin{cases} \Pr(\bigcap_{j=1}^{k-1} E_{i,j} < \lambda_2 \bigcap E_{i,k} \ge \lambda_2 | H_0) & k < \frac{D}{2T} \\ \Pr(\bigcap_{j=1}^{\frac{D}{2T}-1} E_{i,j} < \lambda_2 | H_0) & k = \frac{D}{2T} \end{cases}$$

Accordingly, if the target is absent, the expected number of moves, $\mathcal{E}_0^i(L_i)$, is given by:

$$\begin{aligned} \mathcal{E}_0^i(L_i) &= \mathbb{E}[l_i|H_0] = 1 \cdot \Pr(E_{i,1} \ge \lambda_2 | H_0) \\ &+ \sum_{k=2}^{L_i-1} k \cdot \Pr(\bigcap_{j=1}^{k-1} E_{i,j} < \lambda_2 \bigcap E_{i,k} \ge \lambda_2 | H_0) + L_i \cdot \Pr(\bigcap_{j=1}^{L_i-1} E_{i,j} < \lambda_2 | H_0) \\ &= 1 + \sum_{k=1}^{L_i-1} \Pr(\bigcap_{j=1}^{k} E_{i,j} < \lambda_2 | H_0) \end{aligned} \tag{19}$$

Similarly, if the target is present, the expected number of moves, $\mathcal{E}_1^i(L_i)$, can be derived as:

$$\mathcal{E}_1^i(L_i) = \mathbb{E}[l_i|H_1] = 1 + \sum_{k=1}^{L_i-1} \Pr(\bigcap_{j=1}^{k} E_{i,j} < \lambda_2 | H_1) \tag{20}$$

Suppose there are $M$ mobile sensors in the cluster. The total expected number of moves, $\mathcal{L}_0(\lambda_2, \mathbf{S})$ and $\mathcal{L}_1(\lambda_2, \mathbf{S})$, which are needed to compute the cost defined by (4), are given by:

$$\mathcal{L}_r(\lambda_2, \mathbf{S}) = \sum_{i=1}^{M} \mathcal{E}_r^i(L_i), \quad r = 0, 1 \tag{21}$$

## VI. Near-optimal Sensor Movement Scheduling

In this section, we first analyze the structure of the optimal solution in Section VI-A. A dynamic programming based near-optimal movement scheduling algorithm is proposed in Section VI-B. The procedure of finding the detection thresholds is described in Section VI-C.

### A. The Structure of Optimal Solution

A naive method to solve the problem formulated in Section IV-D is to exhaustively search all possible combinations of $\lambda_1$, $\lambda_2$ and $\mathbf{S}$. Since both $\lambda_1$ and $\lambda_2$ have $k$ possible values and each mobile sensor has $2^{\frac{D}{2T}}$ possible movement schedules, the size of searching space is $k^2 \cdot 2^{\frac{D}{2T} \cdot M}$. Obviously, such exponential complexity is not practical. In this section, we analyze the structure of optimal solution to the problem, which leads to the development of a polynomial-time near-optimal movement scheduling algorithm in Section VI-B.

A solution $< \lambda_1, \lambda_2, \mathbf{S} >$ is said to be *valid* if all constraints can be satisfied. In other words, given a movement schedule $\mathbf{S}$, if $\lambda_1$ and $\lambda_2$ can be found to satisfy the constraints (5) and (6), $< \lambda_1, \lambda_2, \mathbf{S} >$ is a valid solution. A valid solution is *optimal* if it minimizes the objective function (4).

For a movement schedule $\mathbf{X}$, we define $C(\lambda_2, \mathbf{X})$ as the inverse function of $P_{D_2}$ defined by (18):

$$C(\lambda_2, \mathbf{X}) = Q^{-1}(P_{D_2}) = \frac{\frac{n}{2} - \sum_{i=1}^{n} \beta_{2,i}}{\sqrt{\sum_{i=1}^{n} \beta_{2,i} - \sum_{i=1}^{n} \beta_{2,i}^2}} \quad (22)$$

As the local detection probabilities, $\beta_{2,i}$, depend on the detection threshold of the second phase as well as the movement schedule, $C$ is a function of $\lambda_2$ and $\mathbf{X}$. The following theorem reveals the first property of the optimal solution.

*Theorem 1:* Suppose $\mathbf{S}$ and $\mathbf{S}'$ are two valid movement schedules. For a certain $\lambda_2$, if $\mathcal{L}_0(\lambda_2, \mathbf{S}) = \mathcal{L}_0(\lambda_2, \mathbf{S}')$, $\mathcal{L}_1(\lambda_2, \mathbf{S}) = \mathcal{L}_1(\lambda_2, \mathbf{S}')$ and $C(\lambda_2, \mathbf{S}) \leq C(\lambda_2, \mathbf{S}')$, there must exist $\lambda_1$ and $\lambda_1'$, such that $c(\lambda_1, \lambda_2, \mathbf{S}) \leq c(\lambda_1', \lambda_2, \mathbf{S}')$.

*Proof:* Suppose $< \lambda_1, \lambda_2, \mathbf{S} >$ and $< \lambda_1', \lambda_2, \mathbf{S}' >$ minimize the cost function among all valid solutions with schedules $\mathbf{S}$ and $\mathbf{S}'$ for a certain $\lambda_2$, respectively. As $\mathbf{S}$, $\mathbf{S}'$ and $\lambda_2$ are known, such solutions can be found by the exhaustive search of values of $\lambda_1$ in polynomial time. We construct a new solution $< \lambda_1', \lambda_2, \mathbf{S} >$. We now show it is a valid solution. Compared to $< \lambda_1', \lambda_2, \mathbf{S}' >$, this new solution only changes $P_{D_2}$ in all constraints. As $P_{D_2}$ always decreases with $C(\lambda_2, \mathbf{X})$ and $C(\lambda_2, \mathbf{S}) \leq C(\lambda_2, \mathbf{S}')$, we have $P_{D_2}(\lambda_2, \mathbf{S}) \geq P_{D_2}(\lambda_2, \mathbf{S}')$. Therefore, constraint (6) can be met and $< \lambda_1', \lambda_2, \mathbf{S} >$ is a valid solution. Since $\lambda_1$ minimizes the cost function among all valid solutions with $\mathbf{S}$, hence,

$$\begin{aligned} c(\lambda_1,&\lambda_2, \mathbf{S}) \leq c(\lambda_1', \lambda_2, \mathbf{S}) \\ &= (1 - P_a) P_{F_1}(\lambda_1') \mathcal{L}_0(\lambda_2, \mathbf{S}) + P_a P_{D_1}(\lambda_1') \mathcal{L}_1(\lambda_2, \mathbf{S}) \\ &= (1 - P_a) P_{F_1}(\lambda_1') \mathcal{L}_0(\lambda_2, \mathbf{S}') + P_a P_{D_1}(\lambda_1') \mathcal{L}_1(\lambda_2, \mathbf{S}') \\ &= c(\lambda_1', \lambda_2, \mathbf{S}') \end{aligned}$$

■

Theorem 1 shows that, for given total expected numbers of moves, $\mathcal{L}_0$ and $\mathcal{L}_1$, and detection threshold for the second-phase detection, $\lambda_2$, the objective function increases with $C(\lambda_2, \mathbf{X})$. Therefore, the optimal solution $< \lambda_1^*, \lambda_2^*, \mathbf{S}^* >$ must yield the minimum $C(\lambda_2^*, \mathbf{S}^*)$ among all solutions that have the same total expected numbers of moves. Moreover, the maximum total expected numbers of moves for all mobile sensors are bounded. Therefore, for given $\mathcal{L}_0$, $\mathcal{L}_1$ and $\lambda_2$, if there only exist a polynomial number of valid movement schedules, the optimal schedule can be found as the one that minimizes the value of $C(\lambda_2, \mathbf{X})$.

However, according to (19)$\sim$(22), $\lambda_2$, $\mathbf{S}$ and local detection probabilities, $\beta_{2,i}$, have a complex nonlinear relationship, which suggests that there may exist an exponential number of movement schedules for given $\mathcal{L}_0$ and $\mathcal{L}_1$. In the following, we describe a linear approximation of $C(\lambda_2, \mathbf{X})$, which is the key to find the optimal movement schedule in polynomial time.

Denote $\vec{\beta} = (\beta_{2,1}, \ldots, \beta_{2,n})^{\mathrm{T}}$, and $\vec{\varepsilon} = (1, \ldots, 1)^{\mathrm{T}}$, then,

$$C(\lambda_2, \mathbf{X}) = f(\vec{\beta}) = \frac{\frac{n}{2} - \vec{\beta}^{\mathrm{T}} \vec{\varepsilon}}{\sqrt{\vec{\beta}^{\mathrm{T}} \vec{\varepsilon} - \vec{\beta}^{\mathrm{T}} \vec{\beta}}}$$

The first order Taylor expansion of $f(\vec{\beta})$ at $\vec{\beta}_0$ is

$$f(\vec{\beta}) = f(\vec{\beta}_0) + \nabla f(\vec{\beta}_0)^{\mathrm{T}} (\vec{\beta} - \vec{\beta}_0) + R_1 \quad (23)$$

where $\nabla f(\vec{\beta}) = (\frac{\partial f}{\partial \beta_{2,1}}, \ldots, \frac{\partial f}{\partial \beta_{2,n}})^{\mathrm{T}}$, $R_1$ is the remainder and

$$\frac{\partial f}{\partial \beta_{2,i}} = -\frac{1 + \frac{1}{2}(\frac{n}{2} - \vec{\beta}^{\mathrm{T}} \vec{\varepsilon})(\vec{\beta}^{\mathrm{T}} \vec{\varepsilon} - \vec{\beta}^{\mathrm{T}} \vec{\beta})^{-1}(1 - 2\beta_{2,i})}{\sqrt{\vec{\beta}^{\mathrm{T}} \vec{\varepsilon} - \vec{\beta}^{\mathrm{T}} \vec{\beta}}}$$

Since $\beta_{2,i} \in (0, 1)$, if we expand $f(\vec{\beta})$ at the central point, i.e., $\beta_{2,i}^0 = \frac{1}{2}$, we have $\frac{\partial f}{\partial \beta_{2,i}}\Big|_{\beta_{2,i}^0} = -\frac{2}{\sqrt{n}}$ and (23) will be

$$C(\lambda_2, \mathbf{X}) = -\frac{2}{\sqrt{n}} \sum_{i=1}^{n} \beta_{2,i} + \sqrt{n} + R_1 \quad (24)$$

The above equation shows that $C(\lambda_2, \mathbf{X})$ monotonically decreases with $\sum_{i=1}^{n} \beta_{2,i}$, when $R_1$ is independent of $\beta_{2,i}$. Our numerical simulations (omitted due to page limit and can be found in [28]) show that this monotonicity holds with a high probability ($> 98\%$). In practice, $C(\lambda_2, \mathbf{X})$ can be minimized by maximizing $\sum_{i=1}^{n} \beta_{2,i}$. Since the local detection probabilities of fixed sensors are independent of the movement schedule, if the sum of detection probabilities of all mobile sensors is maximized, the sum of detection probabilities of all sensors is also maximized. We now show another property of the optimal solution that further reduces the problem complexity.

*Theorem 2:* Suppose mobile sensor $i$ is scheduled with $L_i$ moves in an optimal schedule. In order to maximize the sum of local detection probabilities in the second-phase detection, the $L_i$ moves must be consecutive from the beginning of the second phase.

The proof of Theorem 2 is omitted here due to page limit and can be found in [28].

### *B. A Near-optimal Movement Scheduling Algorithm*

Based on the analysis on the structure of optimal solution, we develop the following strategy to solve the problem formulated in Section IV-D. First, for given total expected numbers of moves, $\mathcal{L}_0$ and $\mathcal{L}_1$, we employ a dynamic programming algorithm to find the schedule that minimizes $C(\lambda_2, \mathbf{S})$ defined by (22) in polynomial time, which is presented in this section. Then, we search the detection thresholds of two phases, $\lambda_1$ and $\lambda_2$, to find the near-optimal solution in polynomial time, which is presented in Section VI-C. We note that the solution $< \lambda_1, \lambda_2, \mathbf{S} >$ found this way is optimal if the monotonicity between $C(\lambda_2, \mathbf{X})$ and the sum of local detection probabilities holds, i.e., $C(\lambda_2, \mathbf{X})$ strictly decreases with $\sum_{i=1}^{n} \beta_{2,i}$ in (24).

Let $P(m, \mathcal{L}_0^m, \mathcal{L}_1^m)$ be the maximum sum of local detection probabilities of sensors $1, \ldots, m$ with total expected moves no more than $\mathcal{L}_0^m$ and $\mathcal{L}_1^m$ when the target is absent and present, respectively. Then we have a dynamic programming recursion:

$$\begin{aligned} P(m, \mathcal{L}_0^m, \mathcal{L}_1^m) = &\max_{0 \leq L_m \leq H_m} \{ \\ &P(m-1, \mathcal{L}_0^m - \mathcal{E}_0^m(L_m), \mathcal{L}_1^m - \mathcal{E}_1^m(L_m)) + \beta_{2,m}(L_m)\} \end{aligned} \quad (25)$$

where $H_m$ is the maximum number of moves of sensor $m$. $H_m = \min\{\frac{D}{2T}, \frac{x_m^0}{vT}\}$, as the sensor will stop moving if it reaches the surveillance spot or the required delay bound is reached. $\beta_{2,m}(L_m)$ (given by (14)) is the local detection probability of sensor $m$ which is scheduled with consecutive

$L_m$ moves in the second phase. $\mathcal{E}_0^m(L_m)$ and $\mathcal{E}_1^m(L_m)$ are expected numbers of moves of sensor $m$ defined by (19) and (20), if the target is absent and present, respectively. The initial state of the above recursion is $P(0,\cdot,\cdot) = 0$.

According to (25), at the $m^{\text{th}}$ iteration, the optimal value of $P(m, \mathcal{L}_0^m, \mathcal{L}_1^m)$ is computed as the maximum value of $H_m$ cases which have been computed in previous iterations of the recursion. Specifically, for the case where sensor $m$ is scheduled with $L_m$ moves, the sum of local detection probabilities can be computed as $P(m-1, \mathcal{L}_0^m - \mathcal{E}_0^m(L_m), \mathcal{L}_1^m - \mathcal{E}_1^m(L_m)) + \beta_{2,m}(L_m)$ where the first addend is the maximum sum of local detection probabilities for sensor $1, \ldots, m-1$ given the total expected numbers of moves, $\mathcal{L}_0^m - \mathcal{E}_0^m(L_m)$ and $\mathcal{L}_1^m - \mathcal{E}_1^m(L_m)$. According to Theorem 2, sensor $m$'s moves are consecutive from the beginning of the second phase. Therefore, at most $H_m$ cases need to be considered when computing $P(m, \mathcal{L}_0^m, \mathcal{L}_1^m)$. The maximum sum of local detection probabilities for all mobile sensors is given by $P(M, \mathcal{L}_0^M, \mathcal{L}_1^M)$.

In order to calculate $P_{D_2}$ using (18), the square sum of all local detection probabilities is also needed. For each $P(m, \mathcal{L}_0^m, \mathcal{L}_1^m)$, denote $Q(m, \mathcal{L}_0^m, \mathcal{L}_1^m)$ as the square sum of all local detection probabilities initialized to be zero. $Q(m, \mathcal{L}_0^m, \mathcal{L}_1^m)$ is added up incrementally in each iteration. A schedule $\mathbf{S}(m, \mathcal{L}_0^m, \mathcal{L}_1^m)$ is also defined for each $P(m, \mathcal{L}_0^m, \mathcal{L}_1^m)$, and initialized to be empty. $\mathbf{S}(m, \mathcal{L}_0^m, \mathcal{L}_1^m)$ is filled incrementally in each iteration when computing $P(m, \mathcal{L}_0^m, \mathcal{L}_1^m)$. Formally,

$$
\begin{aligned}
L_m^* &= \underset{0 \le L_m \le H_m}{\operatorname{argmax}} \{P(m-1, \mathcal{L}_0^m - \mathcal{E}_0^m(L_m), \mathcal{L}_1^m - \mathcal{E}_1^m(L_m)) + \beta_{2,m}(L_m)\} \\
Q(m, \mathcal{L}_0^m, \mathcal{L}_1^m) &= Q(m-1, \mathcal{L}_0^m - \mathcal{E}_0^m(L_m^*), \mathcal{L}_1^m - \mathcal{E}_1^m(L_m^*)) + \beta_{2,m}^2(L_m^*) \\
\mathbf{S}(m, \mathcal{L}_0^m, \mathcal{L}_1^m) &= \mathbf{S}(m-1, \mathcal{L}_0^m - \mathcal{E}_0^m(L_m^*), \mathcal{L}_1^m - \mathcal{E}_1^m(L_m^*)) \\
&\quad \bigcup \{\mathcal{M}_m(x_m^0 - (j-1)vT, j) \mid 1 \le j \le L_m^*\}
\end{aligned}
$$

Note that both $\mathcal{L}_0$ and $\mathcal{L}_1$ are the expected numbers of moves and hence are real numbers. Their values are discretized in the dynamic programming procedure, i.e., $\mathcal{L}_r = \{0, \Delta, 2\Delta, \ldots, \mathcal{L}_{r,\max}\}, r = 0, 1$, where $\Delta$ is an interval. The maximum value, $\mathcal{L}_{r,\max}$ can be set to be $\max_{\lambda_2 \in \Lambda_2} \sum_{i=1}^{M} \mathcal{E}_r^i(H_i)$, $r = 0, 1$, respectively. The complexity of the dynamic programming procedure is $\mathcal{O}\left(\left(\frac{MD}{\Delta}\right)^2\right)$.

### C. Finding Detection Thresholds

This section presents the procedure of finding the two detection thresholds and the movement schedule. Once the clusters are formed after deployment, $P(M, \mathcal{L}_0, \mathcal{L}_1)$, $Q(M, \mathcal{L}_0, \mathcal{L}_1)$ and $\mathbf{S}(M, \mathcal{L}_0, \mathcal{L}_1)$ are pre-computed for each possible combination of $\lambda_2$ and $< \mathcal{L}_0, \mathcal{L}_1 >$ using the above movement scheduling algorithm:

$$
\begin{aligned}
&\{P_{\lambda_2}(M, \mathcal{L}_0, \mathcal{L}_1) \mid \lambda_2 \in \Lambda_2, \mathcal{L}_0 \in [0, \mathcal{L}_{0,\max}], \mathcal{L}_1 \in [0, \mathcal{L}_{1,\max}]\} \\
&\{Q_{\lambda_2}(M, \mathcal{L}_0, \mathcal{L}_1) \mid \lambda_2 \in \Lambda_2, \mathcal{L}_0 \in [0, \mathcal{L}_{0,\max}], \mathcal{L}_1 \in [0, \mathcal{L}_{1,\max}]\} \\
&\{\mathbf{S}_{\lambda_2}(M, \mathcal{L}_0, \mathcal{L}_1) \mid \lambda_2 \in \Lambda_2, \mathcal{L}_0 \in [0, \mathcal{L}_{0,\max}], \mathcal{L}_1 \in [0, \mathcal{L}_{1,\max}]\}
\end{aligned}
$$

Algorithm 1 shows the pseudo code of the solving procedure. For each possible total expected number of moves, $\mathcal{L}_0$ and $\mathcal{L}_1$, the values of $\lambda_1$ and $\lambda_2$ are searched to minimize the cost defined by (4) under the constraints. A zero cost may occur when all constraints are satisfied without moving the sensors toward the surveillance spot (line 12 in Algorithm 1). We note that the algorithm may not find any valid solution when the performance requirements exceed the maximum detection capability of the cluster. For instance, the constraint on the system detection probability or false alarm rate may not be satisfied even when all mobile sensors have been scheduled with the maximum number of moves under the delay bound. The complexity of Algorithm 1 is $\mathcal{O}\left(\left(\frac{MD}{\Delta}\right)^2 \cdot k^2\right)$.

**Algorithm 1** The procedure of finding the detection thresholds

**Input:** $\Lambda_1$, $\Lambda_2$, $\{P_{\lambda_2}(M, \mathcal{L}_0, \mathcal{L}_1) \mid \lambda_2 \in \Lambda_2, \mathcal{L}_0 \in [0, \mathcal{L}_{0,\max}], \mathcal{L}_1 \in [0, \mathcal{L}_{1,\max}]\}$, $\{Q_{\lambda_2}(M, \mathcal{L}_0, \mathcal{L}_1) \mid \lambda_2 \in \Lambda_2, \mathcal{L}_0 \in [0, \mathcal{L}_{0,\max}], \mathcal{L}_1 \in [0, \mathcal{L}_{1,\max}]\}$, $\{\mathbf{S}_{\lambda_2}(M, \mathcal{L}_0, \mathcal{L}_1) \mid \lambda_2 \in \Lambda_2, \mathcal{L}_0 \in [0, \mathcal{L}_{0,\max}], \mathcal{L}_1 \in [0, \mathcal{L}_{1,\max}]\}$
**Output:** $\lambda_1^*$, $\lambda_2^*$, $\mathbf{S}^*$

```
1:  cost = +∞
2:  for L_0 = [0, Δ, 2Δ, ..., L_{0,max}] do
3:    for L_1 = [0, Δ, 2Δ, ..., L_{1,max}] do
4:      for λ_1 = [λ_{1(1)}, λ_{1(2)}, ..., λ_{1(k)}] do
5:        compute P_{F_1} and P_{D_1} using (10) and (11)
6:        for λ_2 = [λ_{2(1)}, λ_{2(2)}, ..., λ_{2(k)}] do
7:          compute P_{F_2} using (13)
8:          if (5) holds then
9:            compute P_{D_2} using P_{λ_2}(M,L_0,L_1), Q_{λ_2}(M,L_0,L_1) according to (18)
10:           if (6) holds then
11:             compute current cost c using (4)
12:             if c = 0 then
13:               exit
14:             else if c < cost then
15:               cost = c, λ_1^* = λ_1, λ_2^* = λ_2, S^* = S_{λ_2}(M, L_0, L_1)
16:             end if
17:           end if
18:         end if
19:       end for
20:     end for
21:   end for
22: end for
```

## VII. Performance Evaluation

We conduct extensive simulations using the real data traces collected by Duarte et al. [7]. In the experiment, 75 WINS NG 2.0 nodes [29] are deployed to detect military vehicles driving through several intersected roads. The data set used in our simulations includes the time series recorded by 23 nodes at the frequency of 4960Hz and ground truth. Received energy is calculated every 0.75s. Each run is named after the vehicle type and the number of road covered, e.g., AAV3 stands for the data recorded when an Assault Amphibian Vehicle (AAV) drives through the road that is numbered 3. We refer to [7] for more detailed setup of the experiment. In our simulations, the acoustic data of AAV3∼11 are used.

### A. Simulation Methodology

As the data are collected by fixed sensors, they can not be directly used in our simulations. We generate data for our simulations as follows. For each energy measurement collected by a sensor, we compute the distance between the sensor and the vehicle from the ground truth data. When a sensor makes a measurement in our simulations, the energy is set to be the real measurement gathered at a similar distance to target.

While the sensor measurements are directly taken from real data traces, we use a target signal model estimated from a training data set in our movement scheduling algorithm. Such a methodology accounts for realistic factors. For instance, there

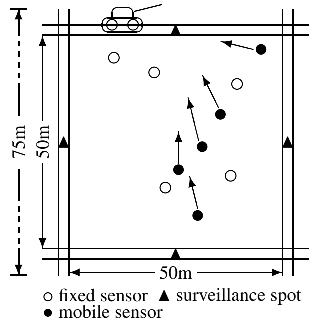



Fig. 3. The deployment region of sensors

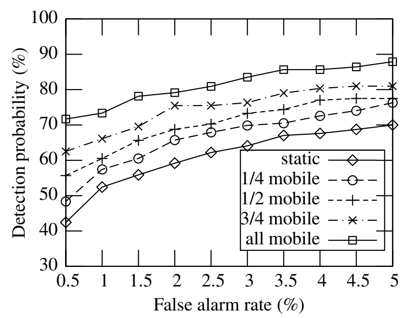



Fig. 4. Receiver operating characteristic (total 12 sensors)

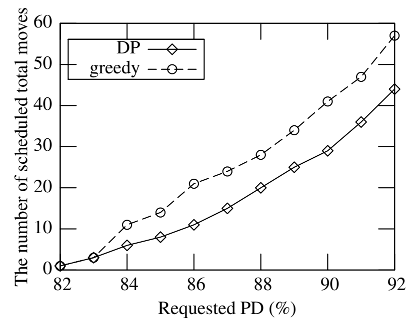



Fig. 5. The number of scheduled total moves vs. requested PD (total 10 mobile sensors)

exists considerable deviation between the measurements of sensors in our simulations and the training data. This deviation is due to various reasons including the difference between vehicles and the changing noise levels caused by wind.

### *B. Simulation Settings*

The simulation code is written in C++. As in [30], we estimate the energy attenuation model using the AAV3 track as the training dataset. Our estimated parameters of the energy model defined by (1) are: $S_0 = 0.51$ (after normalization), $d_0 = 2.6\,\mathrm{m}$, $k = 2$, $\mu = 10^{-4}$, $\sigma^2 = 2\mu$. There are four surveillance spots located at the centers of the road sections. Sensors in our simulations are randomly distributed in a field of $50 \times 50\,\mathrm{m}^2$ surrounded by four road sections, and the length of each road section is $75\,\mathrm{m}$, as illustrated in Figure 3. In real scenarios, surveillance spots would be identified by the network after the deployment. Therefore, it is impossible to deploy sensors only around surveillance spots.

The total simulation time is $3 \times 10^7$ seconds, and each target appearance lasts for 15 seconds. The probability that the target appears at the beginning of a sampling interval is set to be $5\%$. Each sensor in the deployment region thus belongs four clusters. A sensor is randomly selected as the cluster head for each surveillance spot. Based on the local decisions of all sensors in the first-phase detection, a cluster head can determine if a possible target appears at the surveillance spot it monitors. The maximum false alarm rate, $\alpha$, is set to be $5\%$, except in Figure 4. The moving speed of mobile sensors is set to be $1\,\mathrm{m/s}$ except in Section VII-D. Sampling interval, $T$, is set to be $0.75\,\mathrm{s}$, which is consistent to the setting in the SensIT experiment. The search interval in the near-optimal movement scheduling algorithm, $\Delta$, is set to be $0.1$. The expected detection delay, $D$, is set to be $15\,\mathrm{s}$. As our algorithm does not explicitly consider the mobility of targets, we assume that a target remains stationary at a surveillance spot for 15 seconds before it disappears.

### *C. System Detection Performance*

Our first set of simulations evaluate the basic performance of the mobility-assisted detection model and the effectiveness of our movement scheduling algorithm. Figure 4 shows the Receiver Operating Characteristic (ROC) curves for different number of mobile sensors. Under each false alarm rate, the movement schedule of mobile sensors is computed to maximize the system detection probability. Total 12 sensors are deployed. *Static* refers to the deployment in which all sensors remain stationary. *1/4 moible* refers to the 3 mobile sensors and 9 fixed sensors, and so on. We can see that the system detection performance increases significantly with the number of mobile sensors. In particular, six mobile sensors can improve the detection performance by $10\%$ to $35\%$.

In the second set of simulations, we evaluate the effectiveness of our dynamic programming (DP) based movement scheduling algorithm. 10 mobile sensors are deployed. We employ a *greedy* scheduling algorithm as the baseline, in which the cluster head always chooses the mobile sensor closest to the surveillance spot and schedules it with one move until the required detection performance is achieved. Figure 5 shows the total number of moves in the schedules found by different algorithms when the requested detection probability varies from $82\%$ to $92\%$. As shown in Figure 5, our algorithm schedules about 10 fewer moves than the greedy algorithm.

### *D. Impact of Mobile Sensor Speed*

In this set of simulations, we evaluate the impact of mobile sensor speed on the system detection performance. Total 10 mobile sensors are deployed. Figure 6 plots actual detection probability versus the requested detection probability if mobile sensor speed changes from $0.2\,\mathrm{m/s}$ to $1.0\,\mathrm{m/s}$. For each mobile sensor speed, the achievable detection probability yields a saturation point, which occurs when all mobile sensors have moved the maximum distance within the detection delay bound. However, even when the mobile sensors moves as low as $0.2\,\mathrm{m/s}$, a detection probability of $86\%$ is achieved. When mobile sensor speed is higher, the detection probability increases considerably. This result shows that our movement scheduling algorithm can effectively improve system detection performance by taking advantage of the increase of speed.

Figure 7 plots the number of moves versus the requested detection probability. For each speed, the number of moves increases with the requested detection probability. Moreover, for a certain requested detection probability, the total number of moves decreases with sensor speed. It shows that our movement scheduling algorithm can mitigate the impact of low

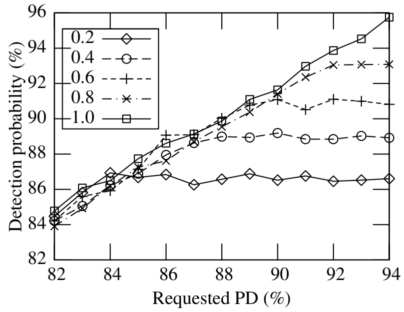



Fig. 6. Actual PD vs. requested PD (total 10 mobile sensors)

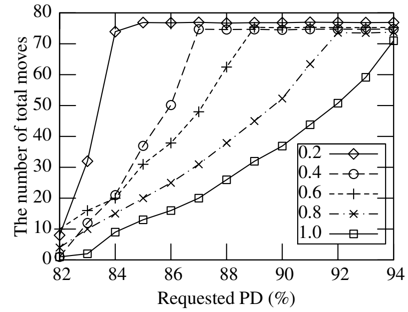



Fig. 7. The number of total moves vs. Requested PD (total 10 mobile sensors)

sensor speed on detection performance by scheduling more sensors to move longer distance toward the target.

## VIII. Conclusion

This paper exploits reactive mobility to improve the detection performance of WSNs. We propose a two-phase detection model in which mobile sensors collaborate with static sensors and move reactively to achieve the required detection performance. We develop a sensor movement scheduling algorithm that can achieve near-optimal system detection performance under given detection delay bounds. Our extensive simulations based on real data traces show that a small number of mobile sensors can significantly improve the system detection performance. Moreover, our movement scheduling algorithm achieves satisfactory performance under realistic settings such as slow speed of sensors (as low as $0.2\,\mathrm{m/s}$).

## Acknowledgment

The work described in this paper was partially supported by the Research Grants Council of Hong Kong under grants RGC 9041266 and 9041258.

## References

[1] D. Li, K. Wong, Y. H. Hu, and A. Sayeed, "Detection, classification and tracking of targets in distributed sensor networks," *IEEE Signal Processing Magazine*, vol. 19 (2), 2002.

[2] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration for tracking applications," *IEEE Signal Processing Magazine*, March 2002.

[3] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-efficient surveillance system using wireless sensor networks," in *MobiSys*, 2004.

[4] R. Pon, M. A. Batalin, J. Gordon, A. Kansal, D. Liu, M. Rahimi, L. Shirachi, Y. Yu, M. Hansen, W. J. Kaiser, M. Srivastava, G. Sukhatme, and D. Estrin, "Networked infomechanical systems: a mobile embedded networked sensor platform," in *IPSN*, 2005.

[5] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling with dynamic deadlines," *IEEE Transactions on Mobile Computing*, vol. 6, no. 4, 2007.

[6] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, "Robomote: enabling mobility in sensor networks," in *IPSN*, 2005.

[7] M. F. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," *Journal of Parallel and Distributed Computing*, July 2004.

[8] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, 2006.

[9] S. Chellappan, W. Gu, X. Bai, D. Xuan, B. Ma, and K. Zhang, "Deploying wireless sensor networks under limited mobility constraints," *IEEE Transactions on Mobile Computing*, vol. 6, no. 10, 2007.

[10] W. W. V. Srinivasan and K.-C. Chua, "Trade-offs between mobility and density for coverage in wireless sensor networks," in *MobiCom*, 2007.

[11] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks," in *MobiHoc*, 2005.

[12] T.-L. Chin, P. Ramanathan, and K. K. Saluja, "Analytic modeling of detection latency in mobile sensor networks," in *IPSN*, 2006.

[13] N. Bisnik, A. Abouzeid, and V. Isler, "Stochastic event capture using mobile sensors subject to a quality metric," in *MobiCom*, 2006.

[14] T.-L. Chin, P. Ramanathan, K. K. Saluja, and K.-C. Wang, "Exposure for collaborative detection using mobile sensor networks," in *MASS*, 2005.

[15] M. A. Batalin, M. Rahimi, Y. Yu, D. Liu, A. Kansal, G. S. Sukhatme, W. J. Kaiser, M. Hansen, G. J. Pottie, M. Srivastava, and D. Estrin, "Call and response: experiments in sampling the environment," in *SenSys*, 2004.

[16] M. Rahimi, M. Hansen, W. J. Kaiser, G. S. Sukhatme, and D. Estrin, "Adaptive sampling for environmental field estimation using robotic sensors," in *IROS*, 2005.

[17] P. Varshney, *Distributed Detection and Data Fusion*. New York, NY: Spinger-Verlag, 1996.

[18] M. Duarte and Y.-H. Hu, "Distance based decision fusion in a distributed wireless sensor network," *Telecommunication Systems*, vol. 26, no. 2-4, 2004.

[19] S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise," in *FUSION*, 2002.

[20] J. Liu, J. Reich, P. Cheung, and F. Zhao, "Distributed group management for track initiation and maintenance in target localization applications," in *IPSN*, 2003.

[21] W.-P. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, 2004.

[22] T. Clouqueur, K. K. Saluja, and P. Ramanathan, "Fault tolerance in collaborative sensor networks for target detection," *IEEE Transactions on Computers*, vol. 53, no. 3, 2004.

[23] H. Liu, X. Jia, P. Wan, C. Yi, S. Makki, and N. Pissinou, "Maximizing lifetime of sensor surveillance systems," *IEEE/ACM Trans on Networking*, vol. 15, no. 2, pp. 334–345, 2007.

[24] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *Infocom*, 2005.

[25] R. Niu, P. K. Varshney, M. Moore, and D. Klamer, "Decision fusion in a wireless sensor network with a large number of sensors," in *FUSION*, June 2004.

[26] I. Olkin, L. J. Gleser, and C. Derman, *Probability Models and Applications*. New York: Macmillan Publishing, 1980.

[27] R. B. Ash and C. A. Doléans-Dade, *Probability & Measure Theory*, 2nd ed. A Harcourt Science and Technology Company, 1999.

[28] R. Tan, G. Xing, J. Wang, and H. C. So, "Collaborative target detection in wireless sensor networks with reactive mobility," City University of Hong Kong, Tech. Rep., 2007, http://www.cs.cityu.edu.hk/~tanrui/pub/mobiddf.pdf.

[29] W. Merrill, K. Sohrabi, L. Girod, J. Elson, F. Newberg, and W. Kaiser, "Open standard development platforms for distributed sensor networks," in *Proceedings of SPIE - Unattended Ground Sensor Technologies and Applications IV 4743*, 2002.

[30] X. Sheng and Y.-H. Hu, "Energy based acoustic source localization," in *IPSN*, April 2003.