

Mobility and Intruder Prior Information Improving the Barrier Coverage of Sparse Sensor Networks

Shibo He, *Member, IEEE*, Jiming Chen, *Senior Member, IEEE*, Xu Li,
Xuemin (Sherman) Shen, *Fellow, IEEE*, and Youxian Sun

Abstract—Barrier coverage problem in emerging mobile sensor networks has been an interesting research issue due to many related real-life applications. Existing solutions are mainly concerned with deciding one-time movement for individual sensors to construct as many barriers as possible, which may not be suitable when there are no sufficient sensors to form a single barrier. In this paper, we aim to achieve barrier coverage in sensor scarcity scenario by dynamic sensor patrolling. In specific, we design a periodic monitoring scheduling (PMS) algorithm in which each point along the barrier line is monitored periodically by mobile sensors. Based on the insight from PMS, we then propose a coordinated sensor patrolling (CSP) algorithm to further improve the barrier coverage, where each sensor's current movement strategy is derived from the information of intruder arrivals in the past. By jointly exploiting sensor mobility and intruder arrival information, CSP is able to significantly enhance barrier coverage. We prove that the total distance that sensors move during each time slot in CSP is the minimum. Considering the decentralized nature of mobile sensor networks, we further introduce two distributed versions of CSP: S-DCSP and G-DCSP. We study the scenario where sensors are moving on two barriers and propose two heuristic algorithms to guide the movement of sensors. Finally, we generalize our results to work for different intruder arrival models. Through extensive simulations, we demonstrate that the proposed algorithms have desired barrier coverage performances.

Index Terms—Mobile Sensor Networks; Barrier Coverage; Periodic Monitoring Scheduling; Coordination Sensor Patrolling; Distributed algorithms

I. INTRODUCTION

Wireless sensor networks (WSNs) are widely recognized as effective surveillance tools for various applications [2]–[8]. Due to real operational limitations such as human inaccessibility, sensors are usually deployed randomly, e.g., dropped by an airplane, in or near a region of interest (ROI). Random sensor dropping causes the WSNs to have topological weaknesses

Part of this paper was presented in 2012 IEEE Infocom conference in Orlando, FL, USA [1].

S. He, J. Chen (corresponding author), and Y. Sun are with the State Key Laboratory of Industrial Control Technology, Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China. S. He was also a visiting scholar at the University of Waterloo, Canada. (email: ferrer@zju.edu.cn; {jmchen, yxsun}@iipc.zju.edu.cn).

X. Li is with INRIA Lille - Nord Europe, Univ Lille Nord de France, USTL, CNRS UMR 8022, LIFL, France (email: xu.li@inria.fr).

X. Shen is with Department of Electrical and Computer Engineering, University of Waterloo, Canada (email: xshen@bbr.uwaterloo.ca).

Research was supported in part by the Natural Science Foundation of China under Grants 61061130563, 61028007, 61004060, 60974122 and 61222305, the Specialized Research Fund for the Doctoral Program of China under Grant 20100101110066 and 20120101110139, the Natural Science Foundation of Zhejiang Province under Grant R1100324, Zhejiang Innovation Program for Graduates under grant YK2010006, and NCET-11-0445.

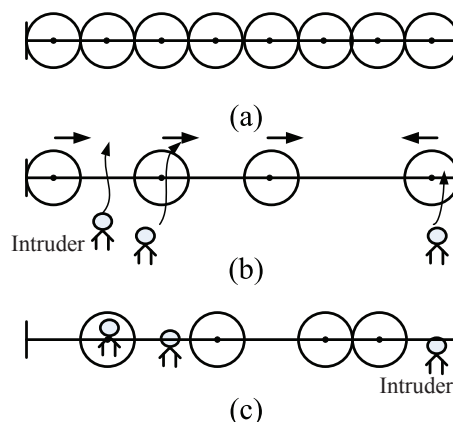


Fig. 1. (a) Full barrier coverage by static sensors; (b) and (c) Barrier coverage by mobile sensors.

such as sensing holes, communication bottlenecks and network partitions. Mobile sensors (e.g., Packbot [9] and Khepera [10]), integrating advanced robotics and sensing technologies, have recently developed to overcome these drawbacks. Unlike traditional static sensors, they have locomotion and are thus able to autonomously improve network performance by adjusting their initial positions to desired ones [11]–[13].

In this paper, we consider a particular scenario, where sensors are not designated to monitor events inside the ROI but to detect intruders that attempt to penetrate the ROI. A real-life example is to deploy sensors on the boundary of a country's territory to identify and prevent illegal entrance to the country. Because sensors are placed within a thin belt region along the ROI boundary acting like a barrier to intruders, the coverage provided by them is referred to as *barrier coverage* [14]–[17]. Existing solutions to barrier coverage in mobile sensor networks implicitly assume the availability of sufficient sensors. They focus on how to move the available sensors one time to construct as many barriers as possible with a minimum aggregate moving distance [18], [19]. These solutions fail to work if a single barrier can not be formed no matter how the sensors are moved due to sensor scarcity. This situation is very likely in reality for budget limitation as it is costly to equip a large number of sensors with locomotion. For example, it may be economically impractical to afford mobile sensors in order to provide full barrier coverage on the boundary of a country's territory. Therefore, it is highly desirable in practice to design a cost-effective barrier coverage, i.e., using mobile sensors as few as possible to meet the requirements.

Fig. 1 shows an example of straight-line barrier coverage. Eight sensors are needed to form a complete barrier according to Fig. 1(a). If only four sensors are available as depicted in

Fig. 1(b), a complete barrier can not be formed by moving each sensor only once. This is illustrated in Fig. 1(c) where the four sensors reach their final positions by the one-time movement indicated by the arrowed lines in Fig. 1(b), with inevitable coverage holes that render some intruders undetected. To improve the barrier coverage performance, it would be better to let mobile sensors patrol along the line dynamically, so that each sensor can be present for intruder detection at different locations with different time.

However, it is a challenging task to design sensor patrolling algorithms for achieving desirable barrier coverage performance. From Fig. 1(b) and Fig. 1(c), there are three intruders trying to cross the barrier line, and sensors have no idea about their arrivals and trajectories. If the sensors move in the directions displayed in Fig. 1(b), only one intruder can be detected (see Fig. 1(c)). In this example, the intruder detection performance will actually be better if they do not move (two intruders rather than one will be detected). This implies if sensors do not know whether their movement will increase the chance of detecting intruders, they probably should stay rather than move blindly. Therefore, the mobility of each sensor has to be carefully controlled in order to effectively increase barrier coverage. The above example indicates the importance of taking into account intruder arrival information for sensor movement scheduling, which is the motivation of this research work.

We consider the barrier coverage problem where m sensors are needed to guarantee full barrier coverage and there are only n mobile sensors available ($n < m$). We first model the arrival of intruders at a specific location as a renew process, in which the next intruder's arrival time is correlated with the current one. The barrier coverage performance is characterized by average intruder detection probability. We formulate the problem as a dynamic programming problem where the movement strategy of all sensors should be made in each time slot dynamically to maximize the intruder detection probability, based on current locations of sensors and intruder arrival information collected in the past time slots. We propose two sensor patrolling algorithms to solve this problem: periodic monitoring scheduling (PMS) and coordinated sensor patrolling (CSP). In PMS, each point of interest in the barrier line is periodically monitored by sensors n times every m time slots, while in CSP the probability of intruder arrival at each point is calculated dynamically, and a coordinated movement strategy is derived accordingly. We then generalize our results to work for other intruder arrival models such as Markov chain. Our main contributions in this paper are as follows.

- We analyze the average intruder detection probability and average sensor moving distance in PMS. We find in PMS the best strategy is to let sensors stay stationary at n fixed points. This conclusion confirms the importance of intruder arrival information for sensor mobility scheduling to improve barrier coverage, and inspires the design of CSP.
- We determine the number of mobile sensors required to guarantee a predefined average intruder detection probability in CSP. We prove that the average per sensor moving distance in each time slot is the minimum.

As CSP is a centralized algorithm, we present its two distributed variants: S-DCSP and G-DCSP.

- We discuss the barrier coverage under scenario where sensors are moving on two barriers. We design two algorithms, I-CSP and J-CSP, to guide the sensor movement in order to detect intruders as many as possible. We generalize our results to work for different intruder arrival models.

The remainder of the paper is organized as follows. We give a brief discussion about the literatures of barrier coverage in Sec. II. We formulate the problem in Sec. III and present PMS along with its performance analysis in Sec. IV. With the insight gaining from PMS, we propose CSP in Sec. V and its two distributed variants S-DCSP and G-DCSP in Sec. VI. We study two-barriers scenario and propose two algorithms in Sec. VI-C. We generalize our results in Sec. VII. Simulation-based performance evaluation is presented in Sec. VIII, followed by the closing remarks in Sec. IX.

II. RELATED WORK

In this section, we introduce the recent results on barrier coverage. Please refer to [20]–[25] for results on other types of coverage. S. Kumar *et al.* [14] introduced the concept of barrier coverage. They defined the notion of *k*-barrier coverage, and proposed algorithms to decide whether a belt region is *k*-barrier covered or not after sensor deployment. They also introduced two probabilistic barrier coverage concepts: *weak barrier coverage* and *strong barrier coverage*. The minimum number of sensors required to ensure weak barrier coverage with high probability has been derived, while the issue of strong barrier coverage is still open.

The barrier coverage problem is very difficult to solve in a decentralized way due to its globalized nature. Chen *et al.* [26] addressed this challenge by introducing the concept of local barrier coverage. Although local barrier is not equivalent to global barrier in general, they showed that it does approximate global barrier in some cases like extremely thin belt regions. Liu *et al.* [15] proposed a distributed algorithm to construct multiple disjoint barriers for strong barrier coverage when sensors are distributed according to Poisson point process. The results hold for any thin belt area of irregular shape, and have the advantages of reduced delay and communication overhead compared with a centralized solution. Chen *et al.* [27] investigated the quality of barrier coverage. Their work can identify when the barrier performance is less than a predefined value and where a repair is needed. Saipulla *et al.* [28] studied the barrier coverage problem when sensors are deployed along a line. The tight lower-bounded probability of the existence of barrier coverage was derived. Yang and Qiao [29] studied the weak barrier coverage by exploiting the sensing collaboration between sensors.

In mobile sensor networks, node mobility has been exploited for autonomous barrier coverage formation and improvement. Saipulla *et al.* [19] studied how to relocate sensors with limited mobility to improve barrier coverage after random sensor deployment. They investigated the effects of the density and mobility of sensors on the barrier coverage improvement, and

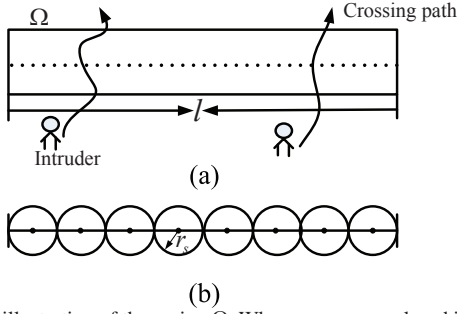


Fig. 2. An illustration of the region Ω . When sensors are placed in the optimal locations (see Fig. 2(b)) along the dash line, Ω can be barrier covered.

proposed an algorithm to check the existence of barrier coverage. Shen *et al.* [18] studied energy-efficient sensor relocation. A centralized algorithm was proposed to compute the optimal positions for all sensors to form a barrier coverage, provided that the initial positions of the sensors are known as a prior. Bhattacharya *et al.* [30] addressed how to optimally move sensors to the boundary of the ROI to form a barrier coverage. Keung *et al.* [16] focused on providing k-barrier coverage against moving intruders. They adopted the classical kinetic theory of gas molecules to analyze the inherent relationship between barrier coverage performance and a set of network parameters such as sensor density and intruder mobility. Bisnik *et al.* [24] considered a scenario where stochastic events arrive at a collection of discrete points along a closed curve, and investigated how the event staying time impacts the event capture performance. They did not consider the temporal correlation between events.

All aforementioned works are concerned with the situations where there are sufficient sensors to build at least one complete barrier coverage. They may not work in the case of sensor scarcity. In this paper, we take the first step to improve barrier coverage in the sensor scarcity situation by letting sensors collaborate with each other to wisely schedule their visits to all points according to the temporal correlation between intruder arrival times. Our work offers a radically new cost-effective barrier coverage solution.

III. PROBLEM FORMULATION

We consider a belt region of interest Ω with two long parallel boundaries. Without loss of generality, let Ω be a rectangle of length l , as illustrated in Fig. 2(a). Intruders may attempt to cross Ω from one boundary to reach the other. Mobile sensors are used to detect intruders.

An intruder is detected by a sensor when the distance between them is less than the sensing range r_s . When traveling around, two sensors can communicate with each other as long as the distance between them is no greater than the communication range r_c . The perfect disc models of sensing and communication are used for ease of presentation. The results presented in the rest of the paper can be easily extended to other complex models, e.g., [31].

Given that Ω is known, the optimal sensor locations for barrier coverage can be pre-calculated according to the existing work on deterministic deployment [14], and sensors only need to move to those deployment points. The optimal sensor locations are the points equally spaced with a distance $2r_s$

on a barrier line (see an illustration in Fig. 2(b)). Denote the number of optimal deployment points by m and the number of available mobile sensors by n . When $n \geq m$, the problem is trivial and has been extensively investigated (see Sec. II for a discussion). We therefore focus on the case of $n < m$.

The operation time of the mobile sensor network is divided into time slots of equal length. As there are not sufficient sensors, at each time slot sensors have to patrol among the m points dynamically so as to enhance the overall barrier coverage performance. At the beginning of each time slot, n points are selected for sensors to monitor; sensors then move to these points and stay there for the rest of the time slot. We assume the time required for decision making and movement is very short and negligible.

Intruders are assumed to arrive stochastically at each point $j, j = 1, 2, \dots, m$ (precisely, in the circle of radius r_s centered at j). At any point j , the intruder interarrival time x is a random variable with a distribution of cumulative function $F(x)$. In many application scenarios, there is temporal correlation between intruder arrival times [32], e.g., when an intruder arrives, the probability that an intruder arrives again in the next few time slots becomes small. Weibull distribution well characterizes this temporal correlation of the intruder arrival time, and has been widely adopted to model many real world random events [33]. The density $f(x)$ and cumulative $F(x)$ functions of a Weibull distribution are given by

$$f(x) = \frac{\beta}{\lambda} \left(\frac{x}{\lambda}\right)^{\beta-1} e^{-\left(\frac{x}{\lambda}\right)^\beta}, \quad (1)$$

$$F(x) = 1 - e^{-\left(\frac{x}{\lambda}\right)^\beta}, \quad (2)$$

where $x \geq 0$, $\lambda > 0$, and $\beta \geq 1$. Note that when $\beta = 1$, Weibull distribution becomes the well-known Poisson distribution. Suppose that the intruder interarrival times are independent and identically distributed (i.i.d.) and an intruder arrives at time slot τ . The probability p_t that the next intruder arrives at time slot $\tau + t$ depends only on the interarrival time t and is given by $p_t = F(t) - F(t-1)$. For easy presentation, we assume that intruder arrival models at all points follow the same Weibull distribution. We note that our results in this work can be directly applied to the heterogeneous case, i.e., intruder arrival models at different points are different.

Denote by a_j^t the state of intruder arrival. $a_j^t = 1$ if an intruder arrives at point j during time slot t , or $a_j^t = 0$ otherwise. Denote by u_j^t the state of sensor presence at point j . $u_j^t = 1$ if there is at least one sensor staying at j at time slot t , or $u_j^t = 0$ otherwise. We characterize the state of point j at time slot t as $s_j^t = (a_j^t, u_j^t)$. An illustration is plotted in Fig. 3. If an intruder arrives at point j during time slot t

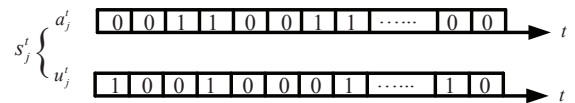


Fig. 3. States of point j at different time slot t .

and a sensor happens to be there at that slot, i.e., $a_j^t = 1$ and $u_j^t = 1$, the intruder is detected. Let L_i^t represent the distance that each sensor $i, i = 1, 2, \dots, n$, moves in time slot t . We define the following two important performance metrics.

Definition 1 (Average intruder detection probability):

Given a sequence of states s_j^t , $j = 1, 2, \dots, m$ and $t = 1, 2, \dots$, the average intruder detection probability γ is defined as

$$\gamma = \lim_{t' \rightarrow \infty} \frac{\sum_{t=1}^{t'} \sum_{j=1}^m a_j^t u_j^t}{\sum_{t=1}^{t'} \sum_{j=1}^m a_j^t}.$$

Definition 2 (Average sensor moving distance): Given

a sequence of moving distances L_i^t , $i = 1, 2, \dots, n$ and $t = 1, 2, \dots$, the average sensor moving distance $\mathcal{L}$ is defined as

$$\mathcal{L} = \lim_{t' \rightarrow \infty} \frac{\sum_{t=1}^{t'} \sum_{i=1}^n L_i^t}{t' \times n}.$$

With these two definitions, the barrier coverage problem can be formulated as how to move n mobile sensors to monitor m points dynamically so as to maximize γ and meanwhile minimize $\mathcal{L}$, i.e.,

$$\begin{aligned} & \max \gamma \text{ while } \min \mathcal{L} \\ \text{s.t. } & \begin{cases} \sum_{j=1}^m u_j^t = n, t = 1, 2, \dots \\ u_j^t = 0 \text{ or } 1, j = 1, 2, \dots, m, t = 1, 2, \dots \end{cases} \end{aligned} \quad (3)$$

Because we measure barrier coverage performance by average intruder detection probability γ , we will use them interchangeably without ambiguity. To ease the presentation, we will also use “monitor” and “occupy” interchangeably to indicate that a sensor is located at a point.

IV. PERIODIC MONITORING SCHEDULING

In this section, we present a periodic monitoring scheduling (PMS) algorithm to solve the barrier coverage problem formulated in Sec. III. PMS is easy to implement and is featured with absence of coordination among sensors.

Recall that there are m points, and we only have $n, n < m$ mobile sensors to monitor these points. During each time slot, there will be $m - n$ points that are not monitored by any sensor. The basic idea of PMS is to let sensors monitor points periodically. Let T denote the number of continuous time slots that a sensor will stay after it reaches another point. In PMS, initially a designated sensor moves to point j , $j = 1, 2, \dots, n$ and stays there for T time slots. Afterwards, the sensor at point j moves to point $\text{mod}(j + n, m)$ and stays there for T time slots. The process continues until all the sensors run out of energy.

Let $m' = \frac{m}{\gcd(m, n)}$, where $\gcd(\cdot)$ is the greatest common divisor function. In PMS, the minimum scheduling period is $m'T$. During every $m'T$ time slots, each point j is monitored by sensors for $n' = \frac{n}{\gcd(m, n)}$ time slots. The ratio of the number of time slots during which there is a sensor monitoring point j to the total number of network operation time slots is therefore $\frac{n}{m}$. An illustration of PMS ($m = 5$, $n = 3$ and $T = 1$) is shown in Fig. 4. The algorithm is sketched in Algorithm 1.

Theorem 1: In PMS, $\gamma = \frac{n}{m}$ and $\mathcal{L} = \frac{2r_s(mn' + nm' - 2nn')}{m'T}$.

Algorithm 1 Periodic monitoring scheduling (PMS)

- 1) Initially:
Assign each sensor a unique ID $i \in [1, n]$
Assign each point a unique ID $j \in [1, m]$
- 2) At time slot $t = 0$:
Let sensor i moves to point i
- 3) At time slot $t = t + T$:
For every point j , the sensor at point j moves to point j_t , where
 $j_t = \text{Mod}(j + n, m)$.
- 4) Terminate if all sensors run out of energy

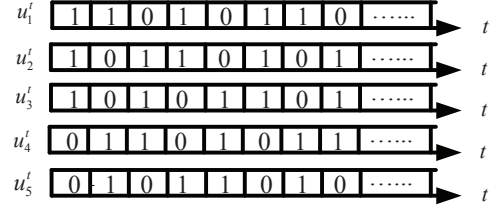


Fig. 4. An illustration of the periodic monitoring scheduling algorithm.

Proof: In PMS, each point is periodically monitored by sensors regardless of the intruder arrival. This is equivalent to the case where sensors have no prior knowledge about intruders. Denote the steady-state probability of intruder arrival at each slot by $\bar{p}$. γ can be calculated as $\gamma = \frac{n\bar{p}}{m\bar{p}} = \frac{n}{m}$.

According to PMS algorithm, sensor at point j will move $2r_s n$ distance to another point j' , $j' = \text{mod}(j + n, m)$, when $j + n \leq m$, and $2r_s(m - n)$ distance when $j + n > m$. For every $m'T$ time slots, a sensor will move $2r_s(m - n)$ distance for n' times and $2r_s n$ distance for $m' - n'$ times. Therefore, the average sensor moving distance $\mathcal{L}$ is

$$\begin{aligned} \mathcal{L} &= \frac{n' \times 2r_s(m - n) + (m' - n') \times 2r_s n}{m'T} \\ &= \frac{2r_s(mn' + nm' - 2nn')}{m'T}, \end{aligned}$$

which completes the proof. ■

Remarks. From the proof of Theorem 1, γ remains the same no matter how sensors are moved. When T goes to infinity, $\mathcal{L}$ approaches zero. Notice that T does not have impact on γ . This indicates that it is better to let sensors stay at n fixed points and leave the remaining $m - n$ points never monitored, when no intruder arrival information is available.

V. COORDINATED SENSOR PATROLLING

In this section, we propose a centralized coordinated sensor patrolling (CSP) algorithm by exploiting the temporal correlation of intruder arrival times to improve average intruder detection probability γ . Its two distributed variants will be introduced in the next section.

A. Preliminaries

To improve γ , the points with high probability of intruder arrival should be selected for sensors to monitor at each time slot. Thus we start with intruder arrival analysis.

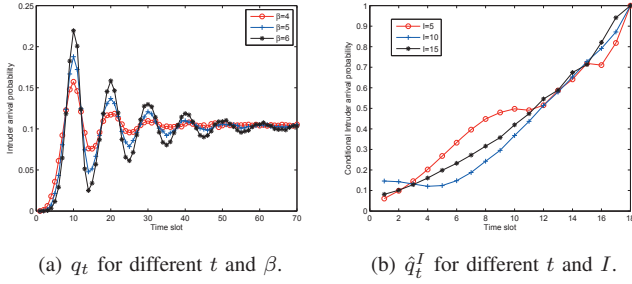


Fig. 5. The intruder arrival probabilities and conditional intruder arrival probabilities at different slots for different values of β .

Theorem 2: When the intruder arrival times are i.i.d. with a cumulative function of $F(\cdot)$, the probability q_t that an intruder arrives at time slot $\tau + t$ is

$$q_t = p_t + \sum_{k=2}^t \sum_{t_1+t_2+\dots+t_k=t} p_{t_1} p_{t_2} \dots p_{t_k}, \quad (4)$$

where τ is the last intruder arrival time, $p_t = F(t) - F(t-1)$ and $t_1, \dots, t_k$ are positive integers.

Proof: We can obtain q_t sequentially. When $t = 1$, q_1 is equal to p_1 . When $t = 2$, the probability that an intruder arrives is subject to the following two cases: i) there is only one intruder arrival during slots $\tau+1$ and $\tau+2$, and it arrives at slot $\tau+2$; and ii) there are two intruder arrivals, one arriving at slot $\tau+1$ and the other at slot $\tau+2$. Hence, q_2 is given by $p_2 + p_1^2$. In general, there may be k intruders, $k = 2, \dots, t$, arriving during time interval $[\tau+1, \tau+t]$. The cases that $k-1$ intruders arrive during interval $[\tau+1, \tau+t-1]$ and one intruder arrives at slot $\tau+t$ can be characterized by $t_1+t_2+\dots+t_k = t$, where $t_1, \dots, t_k$ are the number of slots between intruder arrivals. Then, q_t can be computed as

$$q_t = p_t + \sum_{k=2}^t \sum_{t_1+t_2+\dots+t_k=t} p_{t_1} p_{t_2} \dots p_{t_k}, \quad (5)$$

which completes the proof. ■

It is important to note the difference between p_t and q_t , i.e., p_t is the probability that the next intruder arrival is at slot $\tau+t$ given the last intruder arrival time is τ ; whereas, q_t quantifies the probability that there is an intruder arriving at slot $\tau+t$. When $\tau = 0$, the values of q_t are plotted in Fig. 5(a), where β is a model parameter (see Eqn. (1) and (2)). From this figure, we observe the following important phenomena:

- After an intruder arrives at a point, the probability that an intruder will arrive again at the same point in the next few time slots is very small.
- When t is very large, values of q_t will converge to a constant, implying that the probability that an intruder arrives at this time slot is the same as that at different time slots when we do not have intruder information for a long time.

Corollary 1: The probability q_t^I that the first intruder after time $\tau + I$ arrives at time slot $\tau + I + t$ is

$$q_t^I = p_{I+t} + \sum_{k=1}^I \sum_{t_1+t_2+\dots+t_k \leq I} (p_{t_1} p_{t_2} \dots p_{t_k}) p_{I+t-I'},$$

where $I' = t_1 + t_2 + \dots + t_k$.

Corollary 2: When no intruder arrives during $[\tau + I + 1, \tau + I + t - 1]$, the conditional probability $\hat{q}_t^I$ that an intruder arrives at time slot $\tau + I + t$ is $\hat{q}_t^I = \frac{q_t^I}{1 - \sum_{k=1}^{t-1} q_k^I}$.

Notice the difference between Corollary 1 and Corollary 2. The former describes the general probability about an event arriving at slot $\tau + I + t$, while the latter quantifies the conditional probability based on the knowledge of event arrival during $[\tau + I + 1, \tau + I + t - 1]$. The values of $\hat{q}_t^I$ are plotted in Fig. 5(b), from which we have the following observation:

- As the continuous duration of no intruder arrival at a point increases, the probability that an intruder will arrive at the point increases.

This observation together with previous two observations serve as the design basis of CSP, which is to be elaborated in the next subsection. Note that q_t , q_t^I and $\hat{q}_t^I$ are independent on τ .

Notice that the results in Fig. 5(a) and 5(b) are obtained from simulations. From the expressions of q_t and $\hat{q}_t^I$, we see that there is an exponentially increasing number of possibilities of $t_1, t_2, \dots, t_k$ as t or I grows. This computational complexity makes it difficult for providing numerical results. Since the analysis is easy to follow, it is also not necessary to verify the theoretical findings by comparing the simulation results with numerical results.

B. The algorithmic details

CSP is executed at the beginning of each time slot to determine the movement strategy for each sensor based on the information collected in the past time slots. It runs in two steps: *point selection step*, deciding which n points to be selected for monitoring at current time slot in order to maximize γ ; and *coordinated movement step*, determining how to move sensors to the selected n points with minimum total moving distance. Below, we will elaborate on the two steps of CSP. Their pseudo codes can be found in Algorithm 2.

According to the three observations while analyzing q_t and $\hat{q}_t^I$ in the Sec. V-A, there are three principles for point selection at the first step in order to yield a high γ :

- 1) A sensor should move to another point if it detects an intruder at the point in the previous time slot.
- 2) The point with highest q_t should be selected if a sensor wants to find a point to monitor.
- 3) A sensor should not leave its current point until it detects an intruder.

By principle (1), a sensor is marked *available* if it detects an intruder at the previous time slot, or *unavailable* otherwise. By principle (3), the points where unavailable sensors are located are selected. Denote the total number of available sensors by $\bar{n}$. When $\bar{n} = 0$, i.e., no sensor is available, the algorithm does nothing at the current slot. Let I_j be the number of continuous time slots during which a point j has not been monitored by any sensor since last sensor visit. $I_j = 0$, if a sensor is currently located at point j . Among the points j with $I_j > 0$, $j = 1, 2, \dots, n$, the $\bar{n}$ points with largest $q_t^{I_j}$ are selected in light of principle (2). Since there are $n - \bar{n}$ unavailable sensors at $n - \bar{n}$ points, n points are selected in total.

Algorithm 2 Coordination Sensor Patrol algorithm (CSP)

- 1) $t = 0$; Set $I_j = 0$, for point $j = 1, 2, \dots, m$. Each mobile sensor randomly selects a point to monitor.
 - 2) $t = t + 1$;
 - a) Set $I_j = I_j + 1$ if there is no sensor monitoring at point j at last slot; and $I_j = 0$, otherwise.
 - b) If a mobile sensor detects an intruder at slot $t - 1$, the mobile sensor claims itself as available. Count the total number of available sensors $\bar{n}$. If $\bar{n} = 0$, let $t = t + 1$, go to Step 2).
 - c) Compute $q_t^{I_j}$ for those points that are not monitored by mobile sensors at slot $t - 1$. The $\bar{n}$ points of largest $q_t^{I_j}$ are newly selected for sensors to monitor. Together with other $n - \bar{n}$ points that are monitored by $n - \bar{n}$ unavailable sensors, n points are selected.
 - d) Given point set $C = \{j_1, j_2, \dots, j_n\}$, and sensor set $C' = \{i_1, i_2, \dots, i_n\}$, move mobile sensor i_k to point j_k for intruder monitoring, $k = 1, 2, \dots, n$.
 - 3) Continue Step 2) until all mobile sensors run out of energy.
-

Let C be the set of points selected at the first step and C' the set of sensors. At the second step, the points in C are sorted as $\{j_1, j_2, \dots, j_n\}$ according to their sequence on the barrier line, from one end to the other, and the sensors in C' are ordered similarly as $\{i_1, i_2, \dots, i_n\}$ according to their locations. The coordinated movement strategy is as follows:

$$i_1 \longrightarrow j_1, i_2 \longrightarrow j_2, \dots, i_n \longrightarrow j_n.$$

According to this strategy, unavailable sensors do not necessarily stay where they were in order to reduce the total moving distance of each sensor. We will prove in the next subsection that CSP is the optimal movement strategy in terms of minimizing $\mathcal{L}$.

C. Performance analysis

1) *Total number of sensors*: In a real-life barrier coverage application, there is often a threshold requirement for the intruder detection probability, i.e., $\gamma \geq \gamma_0$, given system parameters. In the following, we derive the number of sensors needed in order for CSP to meet this requirement.

In CSP, a sensor stays at a point until it detects an intruder at the point. Without loss of generality, let τ be the ending time slot of a sensor monitoring. Denote by $\bar{I}$ the average inter-sensor-monitoring duration, i.e., the average number of continuous time slots that a point is not monitored by any sensor. According to CSP, the detection probability loss only occurs during the $\bar{I}$ time slots after τ . To guarantee $\gamma \geq \gamma_0$ is equivalent to ensure the detection probability loss to be less than $1 - \gamma_0$, i.e.,

$$q_1 + q_2 + \dots + q_{\bar{I}} < 1 - \gamma_0. \quad (6)$$

Given γ_0 , by Theorem 2, we can readily find the $\bar{I}$ that satisfies this Inequality. When there are multiple such $\bar{I}$, we take the largest one.

After finding $\bar{I}$, we can compute the average number $\mathcal{I}$ of time slots that a sensor should stay at a point for continuous monitoring before it leaves for other points. Recall that the intruder detection probability at time slot $\tau + \bar{I} + t$ is $q_t^{\bar{I}}$. Then, $\mathcal{I}$ is the expected value of t and can be calculated as

$$\mathcal{I} = \sum_{t=1}^{\infty} t q_t^{\bar{I}}. \quad (7)$$

The average monitoring ratio (AMR) of the number of time slots when a sensor is occupying (monitoring) a point to the total number of time slots that the network operates is thus given by $\frac{\mathcal{I}}{\bar{I} + \mathcal{I}}$. Obviously, AMR is upper-bounded by $\frac{n}{m}$, where n is the number of sensors and m the number of points. That is,

$$\frac{n}{m} \geq \frac{\mathcal{I}}{\bar{I} + \mathcal{I}}. \quad (8)$$

By solving this inequality, we find the smallest n and then take it as an estimate of the number of mobile sensors required for achieving barrier coverage performance requirement.

Remarks. i) From the above derivation process of n , the barrier coverage performance γ of the resultant mobile sensor network is an approximation of γ_0 . Due to the extreme difficulty in calculating an exact γ , we use the average $\bar{I}$ and $\mathcal{I}$ to give an estimate. Later, through simulation we will show such approximation estimate is efficient; and ii) it is hard, if not impossible, to get explicit expressions for Eqn. 6 and 8. Approximate numerical results should be employed when computing $\bar{I}$ and $\mathcal{I}$.

2) *Average sensor moving distance*: Recall that at the coordinated movement step of CSP, we sort the selected deployment points as $j_1, j_2, \dots, j_m$, and the sensors as $i_1, i_2, \dots, i_n$ according to their locations. To simplify the presentation, we also use i_k to denote the location of sensor i_k for $k = 1, 2, \dots, n$ when there is no ambiguity.

Let $i'_1 i'_2 \dots i'_n$ be a permutation of $i_1 i_2 \dots i_n$. It represents a movement strategy, in which sensor i'_k is scheduled to move to point j_k . Let $L^t(i'_1 i'_2 \dots i'_n)$ denote the total moving distance of all sensors by the strategy at slot t , i.e., $L^t(i'_1 i'_2 \dots i'_n) = \sum_{k=1}^n d_{i'_k j_k}$. To minimize the average sensor moving distance, we need to find a permutation $i'_1 i'_2 \dots i'_n$ such as to minimize $L^t(i'_1 i'_2 \dots i'_n)$. We have the following optimality theorem.

Theorem 3: CSP yields an optimal mobility scheduling solution in terms of minimizing average sensor moving distance at each time slot.

Proof: CSP adopts the movement strategy $i_1 i_2 \dots i_n$ at each time slot. We denote it by G_0 . In order to prove G_0 is optimal, we have to show $L^t(i'_1 i'_2 \dots i'_n) \geq L^t(G_0)$ holds for its any permutation $i'_1 i'_2 \dots i'_n$.

For any two items a and b in a sequence, we define $a < b$ if a precedes b . In G_0 , we then have $i_1 < i_2 < \dots < i_n$. Let $g_{kk'}$, $k < k'$, denote a permutation operation that swaps the k -th and k' -th items in a sequence. The permutation of G_0 by this operation can be expressed as $G_{kk'} = G_0 \circ g_{kk'} = i_1 \dots i_{k-1} i_{k'} i_{k+1} \dots i_{k'-1} i_k i_{k'+1} \dots i_n$. We first show

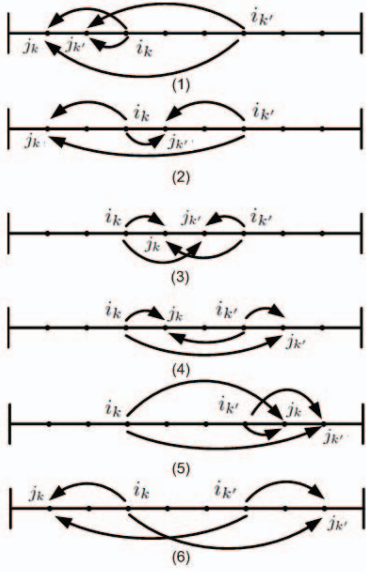


Fig. 6. Six cases about relationships between sensors $j_k, j_{k'}$ and $i_k, i_{k'}$. For each case, sensors i_k and $i_{k'}$ can move to either j_k or $j_{k'}$.

the following inequality holds for any i_k and $i_{k'}$ satisfying $i_k \leq j_{k'}$,

$$L^t(G_{kk'}) \geq L^t(G_0).$$

The moving distance of all sensors but i_k and $i_{k'}$ are the same in the two strategies $G_{kk'}$ and G_0 . We thus only need to consider the moving distances of these two particular sensors. In G_0 , sensor i_k moves to point j_k , and sensor $i_{k'}$ moves to point $j_{k'}$; in $G_{kk'}$, sensor $i_{k'}$ moves to point j_k , and i_k moves to point $j_{k'}$. We denote the moving distance of sensor i_k and $i_{k'}$ in G_0 by $L^t(i_k i_{k'})$ and that in $G_{kk'}$ by $L^t(i_{k'} i_k)$. There are six different cases of the relation between locations of $j_k, j_{k'}$, and $i_k, i_{k'}$. They are illustrated in Fig. 6.

In case 1, points j_k and $j_{k'}$ are both on the left of sensors i_k and $i_{k'}$. The total moving distance of sensors i_k and $i_{k'}$ in G_0 is $L^t(i_k i_{k'}) = d_{i_k j_k} + d_{i_{k'} j_{k'}} = d_{i_k j_{k'}} + d_{j_{k'} j_k} + d_{i_{k'} j_k} = d_{i_k j_{k'}} + d_{i_{k'} j_k} = L^t(i_{k'} i_k)$.

In case 2, point j_k is on the left of sensor i_k (including $j_k = i_k$) and point $j_{k'}$ is between sensors i_k and $i_{k'}$. We compute $L^t(i_k i_{k'}) = d_{i_k j_k} + d_{i_{k'} j_{k'}}$ and $L^t(i_{k'} i_k) = d_{i_k j_{k'}} + d_{i_{k'} j_k}$. Because $d_{i_k j_{k'}} + d_{i_{k'} j_k} = d_{i_{k'} j_{k'}} + 2d_{j_{k'} i_k} + d_{i_k j_k}$, we have $L^t(i_k i_{k'}) < L^t(i_{k'} i_k)$.

In case 3, points j_k and $j_{k'}$ are both between sensors i_k and $i_{k'}$. We may easily have $L^t(i_k i_{k'}) = d_{i_k j_k} + d_{i_{k'} j_{k'}} = d_{i_{k'} j_{k'}} + 2d_{j_{k'} i_k} + d_{i_k j_k} = d_{i_{k'} j_k} + d_{i_k j_{k'}} < L^t(i_{k'} i_k)$. Similarly, we can have the same result $L^t(j_k j_{k'}) \leq L^t(j_{k'} j_k)$ in the other cases. Therefore, we conclude $L^t(G_{kk'}) \geq L^t(G_0)$.

Next, we prove the theorem by induction. Let G denote a movement strategy $i'_1 i'_2 \dots i'_n$. When $n = 2$, we have $L^t(i_1 i_2) \leq L^t(i_2 i_1)$, and thus $L^t(G_0) \leq L^t(G)$. Suppose $L^t(G_0) \leq L^t(G)$ holds in the case $n = N - 1$. We now consider the case $n = N$.

When $i_1 = i'_1$, we know $d_{i_1 j_1} = d_{i'_1 j_1}$. As the conclusion holds when $n = N - 1$, we have $L^t(i_2 i_3 \dots i_n) \leq L^t(i'_2 i'_3 \dots i'_n)$. Hence, $L^t(G_0) \leq L^t(G)$ is true. We next show $L^t(G_0) \leq L^t(G)$ is true when $i_1 \neq i'_1$. Note that G_0 can be transformed into $i'_1 i_2 \dots i_n$ by a finite series of permutation operations $g_{k_1, k_1+1}, g_{k_2, k_2+1}, \dots, g_{k_w, k_w+1}$.

Since $L^t(G_0) \leq L^t(G_0 \circ g_{k, k+1}) \leq \dots \leq L^t(G_0 \circ g_{k_1, k_1+1} \circ \dots \circ g_{k_w, k_w+1}) = L^t(i'_1 i_2 \dots i_n)$, and by induction $L^t(i_2 i_3 \dots i_n) \leq L^t(i'_2 i'_3 \dots i'_n)$ holds, we obtain

$$\begin{aligned} L^t(G_0) &= L^t(i_1 \dots i_n) \leq L^t(i'_1 i_2 \dots i_n) \\ &= d_{i'_1 j_1} + L^t(i_2 i_3 \dots i_n) \\ &\leq d_{i'_1 j_1} + L^t(i'_2 i'_3 \dots i'_n) \\ &= L^t(i'_1 i'_2 \dots i'_n) = L^t(G). \end{aligned}$$

■

VI. DISTRIBUTED CSP (DCSP) AND TWO BARRIERS CASE

In this section, we introduce two distributed CSP algorithms, i.e., S-DCSP and G-DCSP. In particular, the latter is a generalization of the former. We also discuss about the scenario where sensors move on two barriers to monitor intruders.

A. Simple DCSP (S-DCSP)

S-DCSP consists of two phases: i) an initialization phase, and ii) a dynamic movement phase.

1) *Initialization*: In the initialization phase, the sensors are assumed to be connected, and one of them is elected as leader. The leader is responsible for distributing the preference level (initially it equals to 1) of each sensor among the points, indicating how the sensor likes to monitor the points. It first sorts all the points according to their positions along the barrier line. Then it performs preference distribution for all the sensors one by one, in the increasing order (starting from 1-st sensor). For sensor i , $1 \leq i \leq n$ (the i -th sensor), the leader assigns a preference level $0 \leq pl_i^j < 1$ to point j , $1 \leq j \leq m$ (the j -th point) sequentially in the increasing order (starting from 1-st point), subject to the constraint $\sum_{j=1}^m pl_i^j = 1$.

When considering point j , the leader compares the sensor preference $pl_i^j = \sum_{i=1}^n pl_i^j$ aggregated on point j with n/m . Denote by $\hat{pl}_i$ the remaining preference level of sensor i . If $pl_i^j < n/m$, the leader sets $pl_i^j = \min\{\hat{pl}_i, n/m - pl_i^j\}$ and deducts this amount of preference from $\hat{pl}_i$, i.e., $\hat{pl}_i = \hat{pl}_i - pl_i^j$; otherwise, it precedes to consider the next point. As soon as $\hat{pl}_i$ becomes 0, it sets the preference level of sensor i for the rest points to 0 and starts to serve the next sensor. According to this preference distribution method, sensor i will be in favor of nearby points and may have zero preference to points relatively far. For example, when $n = 3$ and $m = 5$, we have the following sensor preference distribution:

- $pl_1^1 = \frac{3}{5}, pl_1^2 = \frac{2}{5}, pl_1^3 = pl_1^4 = pl_1^5 = 0$;
- $pl_2^1 = \frac{1}{5}, pl_2^2 = \frac{3}{5}, pl_2^3 = \frac{1}{5}, pl_2^4 = pl_2^5 = 0$;
- $pl_3^1 = \frac{2}{5}, pl_3^2 = \frac{3}{5}, pl_3^3 = pl_3^4 = pl_3^5 = 0$.

At the end of the initialization phase, sensor i is associated with a point set MS_i , to which it has a non-zero preference level. In the above example, $MS_1 = \{1, 2\}$, $MS_2 = \{2, 3, 4\}$ and $MS_3 = \{4, 5\}$. The leader informs sensor i about MS_i , which are the points that sensor i will move to monitor in the following dynamic movement phase. Notice that i) each point finally has exactly n/m amount of aggregated sensor preference, which lets the algorithm yield an average monitoring ration (AMR) equal or nearly equal to the value n/m ; and ii) sensor i and $i + 1$ may have a common point in

their MS_i and MS_{i+1} (as shown in the previous example), and sensor i and sensor i' , $i' = 1, \dots, i-2, i+2, \dots, m$, do not have a common point to monitor.

2) *Dynamic movement*: In the dynamic movement phase, each sensor i moves between points in MS_i . For each point $j, j \in MS_i$, sensor i maintains the number of time slots, denoted by I_{ij} , for which it has not monitored point j since its last visit. At the beginning of each time slot, sensor i makes decision whether to move and where to move. It will decide to stay at its current point if it did not detect an intruder at the last time slot, or move to another point otherwise. In order to find a new point to move to, sensor i calculates the intruder arrival probability $q_t^{I_{ij}}$ for every point $j, j \in MS_i$. The point with the largest $pl_i^j \times q_t^{I_{ij}}$ is selected. Once the movement destination is determined, it moves immediately and stays there for the current time slot.

Due to independent decision making, collision may occur, i.e., two adjacent sensors i and $i+1$ may select the same point $j, j \in MS_i \cap MS_{i+1}$ for monitoring. If sensor i moves to point j and finds (through location communication) that the point has been monitored by sensor $i+1$ for at least one time slot, it will set $I_{ij} = 0$, recalculate $q_t^{I_{ij}}, j \in MS_i$ and find another point to monitor. If sensor i and $i+1$ both start to monitor point j at the current time slot, they will enter a competition for monitoring j . In the competition, sensor i and sensor $i+1$ generate random numbers from $[0, \frac{pl_i^j}{pl_i^j + pl_{i+1}^j}]$, and $[0, \frac{pl_{i+1}^j}{pl_i^j + pl_{i+1}^j}]$, respectively, and exchange their numbers through local communication; the one with the larger random number wins, and the other has to set $I_{ij} = 0$, and recalculate $q_t^{I_{ij}}$ to find another point for monitoring. The competition is repeated in case of tie.

B. General DCSP (G-DCSP)

We generalize S-DCSP to obtain a new DCSP algorithm, named by G-DCSP. In G-DCSP, sensors are assumed to be clustered. Depending on applications, clustering can be done in different ways. Denote the number of clusters by $\tilde{n}$ and the set of sensors in each cluster k by $SC_k, k = 1, 2, \dots, \tilde{n}$. Let $n' = \tilde{n} - \text{Mod}(\tilde{n}, n)$. The following is a simple clustering method:

$$\begin{aligned} SC_1 &= \{1, 2, \dots, \lfloor \frac{n}{\tilde{n}} \rfloor\}, \\ SC_2 &= \{\lfloor \frac{n}{\tilde{n}} \rfloor + 1, \dots, 2\lfloor \frac{n}{\tilde{n}} \rfloor\}, \\ &\vdots \\ SC_{n'} &= \{(n' - 1)\lfloor \frac{n}{\tilde{n}} \rfloor + 1, \dots, n'\lfloor \frac{n}{\tilde{n}} \rfloor\}, \\ SC_{n'+1} &= \{n'\lfloor \frac{n}{\tilde{n}} \rfloor + 1, \dots, (n' + 1)\lfloor \frac{n}{\tilde{n}} \rfloor + 1\}, \\ &\vdots \\ SC_{\tilde{n}} &= \{m - \lfloor \frac{m}{\tilde{n}} \rfloor, \dots, m\}. \end{aligned}$$

G-DCSP extends the initialization phase of S-DCSP by requiring each cluster SC_k to compute the union $\overline{MS}_k$ of the point sets which are assigned to its member sensors, i.e., $\overline{MS}_k = \cup_{t \in SC_k} MS_t$. This computation can be performed

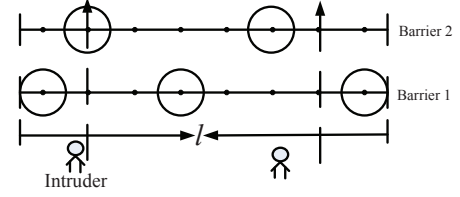


Fig. 7. Network topology under two-barriers case.

by the cluster head of SC_k , which then passes the results to its cluster members. In the dynamic movement phase, at the beginning of each time slot, the sensors in SC_k go to a rendezvous point to fuse their information, find points among $\overline{MS}_k$ to monitor using the centralized CSP algorithm, inform each other about the points that they decide to monitor and then move to their selected points. The rendezvous point is a point that minimizes the total moving distance of the sensors for rendezvous. It is computed by the sensors locally since they know each other's monitoring points in the previous time slot. Monitoring collision is possible as there may be common points in two clusters' point sets. It can be resolved in the same way as in S-DCSP.

Remarks. In S-DCSP, each sensor works independently after the initialization phase and do not rely on the information of other sensors. In G-DCSP, sensors are clustered, and sensors in the same cluster have physically meet and communication in order to make protocol decision. S-DCSP involves less communication and movement cost than G-DCSP, while G-DCSP has a better performance γ than S-DCSP. They should be selected for use according to application-specific requirements.

C. Two-Barriers Case

So far, we have considered the case where mobile sensors are moving along a single barrier to detect intruders. We consider two deployment barriers in this subsection to investigate if the performance of barrier coverage can be improved. Cases of $k, 3 \leq k \leq n$, deployment lines can be solved in a similar, possibly more complex way.

Assume there are two deployment barriers in the region Ω . n_1 sensors are assigned to move along the barrier 1 and n_2 sensors to move along the barrier 2, $n_1 + n_2 = n$. Intruders try to cross the region from one entrance side to the destination side. Without loss of generality, we suppose intruders will first pass through barrier 1 and then barrier 2. Intruders are assumed to arrive at the barrier 1 according to Weibull distribution and if not detected by sensors at the barrier 1, they will proceed to pass through barrier 2 at next slot. As intruders can across the region Ω along an arbitrary path, the intruders appearing at point $j, j = 1, 2, \dots, m$, can appear at any point at barrier 2. We consider a simple case in this section, i.e., intruders choose the shortest path to cross the region and therefore intruder at point $j, j = 1, 2, \dots, m$, will proceed to pass through point j at barrier 2. General case will be included in our future work. Other network settings are the same as those in the single barrier case.

We introduce two sensor movement strategies: i) Independent CSP (I-CSP), in which sensors on two different barriers act independently and use CSP algorithm to guide their movement, and ii) Joint CSP (J-CSP), in which the

information of intruder arrival at the barrier 1 can be exploited for movement strategy design on the barrier 2. We elaborate I-CSP and J-CSP in the following.

I-CSP. Initially, n_1 sensors are placed at points 1 to n_1 on the first barrier, and n_2 sensors are placed at points $m - n_2 + 1$ to m . Sensors at barrier 1 employ the CSP algorithm to guide their movement dynamically. Sensors at barrier 2 do not know the intruder detection at barrier 1, and thus assume that the intruder arrival distribution at each point j is the same as that of the point j on the barrier 1. The CSP algorithm is then used to obtain the monitoring points for sensors on the barrier 2.

J-CSP. The initial position of each sensor in J-CSP is the same as that in I-CSP. Sensors on two barriers jointly exploit the intruder arrival information. Specifically, sensors on the barrier 1 calculate I_j (defined in Sec. V-B) for each point j at each slot t by employing CSP algorithm. n_1 points with largest $q_t^{I_j}$ on the barrier 1 are selected for sensors to monitor at slot t . Then these $q_t^{I_j}$ information are transmitted to sensors on the barrier 2 to calculate the points to be monitored in the next slot. Initially at the first slot, sensors at barrier 2 stay at points $m - n_2 + 1$ to m , as they do not have any information about $q_t^{I_j}$.

Remark. In I-CSP, sensors on barrier 1 and barrier 2 adopts CSP to find monitoring points independently. It is possible that a sensor monitors point j on the barrier 1 at slot $t - 1$ and some sensor stays at point j on the barrier 2 at slot t . The sensor at point j on barrier 2 at slot t is redundant since intruder arriving at point j on barrier 2 at slot t can be detected at the slot $t - 1$ by the sensor at point j on the barrier 1. This will never happen to CSP for the single barrier approach. In J-CSP, the movement of sensors on the barrier 2 depends on the information gathered by sensors on the barrier 1. The more information gathered by sensors on the barrier 1, the better the overall barrier coverage performance will be. Therefore, it is advisable to allocate sensors on the barrier 1 as many as possible. This is also validated by the simulation results in Fig. 18, where γ increases as the number of sensors on the barrier 1 increases. Summarily, CSP has a better barrier coverage performance than I-CSP and J-CSP.

VII. GENERALIZATION: DIFFERENT INTRUDER ARRIVAL MODELS

In this section, we discuss how to generalize our results obtained in the previous sections.

A. Different Intruder Arrival Models

In the previous sections, we adopt the Weibull distribution to model the intruder arrival model. In this subsection, we study how to modify our results to other types of intruder arrival models. We only focus on CSP since other algorithms (S-DCSP, G-DCSP, I-CSP and J-CSP) can be studied in the same way.

In CSP, we try to allocate sensors to monitor the points with high intruder arrival probabilities. There are three cases to calculate the probability of intruder arrival at current slot for each point: i) a sensor was monitoring at the point at last slot and an intruder was detected; ii) a sensor was monitoring

Algorithm 3 Modified Coordinated Sensor Patrolling algorithm (MCSP)

- 1) $t = 0$; Set $I_j = 0$, for point $j = 1, 2, \dots, m$. Each mobile sensor randomly selects a point to monitor.
- 2) $t = t + 1$;
 - a) Set $I_j = I_j + 1$ if there is no sensor monitoring at point j at last slot; $I_j = 0$, otherwise.
 - b) If $I_j > 0$, calculate the intruder arrival probability at point j , $(T^{I_j})_{s1}$. If $I_j = 0$ and a mobile sensor detected an intruder at slot $t - 1$, the intruder arrival probability at point j is T_{11} ; otherwise, the intruder arrival probability at point j is T_{01} .
 - c) Find the n points with highest intruder arrival probabilities. Denote the point set by C .
 - d) Points in set $C = \{j_1, j_2, \dots, j_n\}$ are sorted according to their sequences on the barrier line. Sensors $C' = \{i_1, i_2, \dots, i_n\}$ are similarly sorted according to their locations on the barrier line. Move mobile sensor i_k to point j_k for intruder monitoring, $k = 1, 2, \dots, n$.
- 3) Continue Step 2) until all mobile sensors run out of energy.

at the point at last slot and no intruder was detected; iii) no sensor was allocated to monitor the point. Our results can be modified to scenarios with other types of intruder arrival models as long as we can calculate the probabilities of intruder arrival at current slot under the aforementioned three cases. We use the well-known Markov chain model as an example to illustrate the generalization of our results.

By a Markov chain model, there are two states at each point at each slot, 0 means no intruder arrival and 1 means intruder arrival. The state transition matrix T is given by

$$T = \begin{pmatrix} T_{00} & T_{01} \\ T_{10} & T_{11} \end{pmatrix}. \quad (9)$$

Using T , the probability of intruder arrival at current slot for case i-iii) can be calculated as T_{11} , T_{01} and $(T^I)_{s1}$, respectively, where $I - 1$ is the number of slots since last sensor left, $T^I = \underbrace{T \times T \times \dots \times T}_I$, s is the state detected by

the last sensor before it left and $(T^I)_{s1}$ denotes the transition probability from state s to 1 in transition matrix T^I . We give the modified CSP algorithm for Markov chain model in Algorithm 3.

B. Spatial correlations between points

So far, we have assumed that the intruder arrival times between different points are independent and identically distributed. In scenarios where the intruder arrivals at different points are highly correlated, exploiting such spatial correlations would greatly improve the barrier coverage performance. In this subsection, we discuss how to modify our solutions to incorporate spatial correlations.

Let E_j denote the event that an intruder arrives at current slot at point j , and $\mathbb{P}(E_1, E_2, \dots, E_m)$ is the joint

probability that there is an intruder present at each point $j, j = 1, 2, \dots, m$. In our previously proposed algorithms, we first calculate the intruder arrival probability at each point j , i.e., $\mathbb{P}(E_j)$, and then find the points with highest $\mathbb{P}(E_j)$ for sensors to monitor. Taking Algorithm 3 for an example, when $\mathbb{P}(E_j), j = 1, 2, \dots, m$, are highly correlated, we only have to modify the process of finding the n points with highest probabilities of intruder arrival in Step 2c) in the following way. Initially, find point j_1 with highest $\mathbb{P}(E_{j_1})$. Given $j_1, j_2, \dots, j_{k-1}$, find the j_k such that

$$j_k = \arg \max \mathbb{P}(E_j | E_{j_1} E_{j_2} \dots E_{j_{k-1}}).$$

VIII. SIMULATION RESULTS

In this section, we conduct simulations to validate the analysis and the performance of the proposed algorithms. We use MATLAB to perform our simulations. The network operation time is divided into time slots, each with 1 unit simulated time.

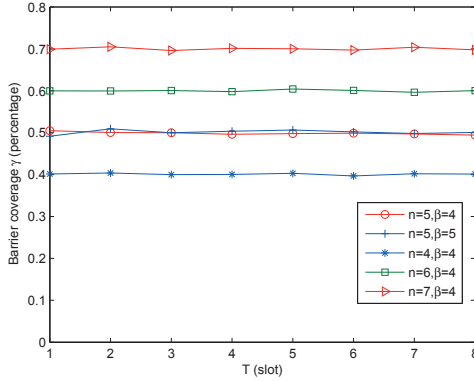


Fig. 8. Performance of γ for different T, n and β , and fixed $m = 10$. Values of γ are equal to n/m for all scenarios.

A. Single Barrier case with intruder arrival following Weibull distribution

Intruders are simulated to arrive from time to time according to i.i.d. Weibull distribution. For all the simulations, $\lambda = 10$ (see Eqns. 1 and 2). An intruder is detected when it arrives at a point and a sensor is monitoring there. The average intruder detection probability γ is calculated by the ratio of the number of detected intruders to all arriving intruders. As there is no existing work on our problem, we use simulations to demonstrate the performance of the proposed algorithms.

We first evaluate the performance of PMS. In the simulation, initially n sensors are located at points $1 \sim n$. The total number of points m is set to be 10. For every T time slots, the sensor at point j will move to point $j_t = \text{Mod}(j + n, m)$ for monitoring, regardless of the arrival of intruder. We first fix $n = 5$ and $\beta = 4$, and vary T to show the impact of T on γ . The results are plotted in Fig. 8. As stated in the section IV, values of γ are equal to n/m for all different T , reflecting the continuous monitoring time T at a point does not impact γ . Then we vary the values of n to 4, 6, 7, and 8, and conduct the corresponding simulations. Values of γ in these cases are still equal to n/m . At last, we set $\beta = 5, n = 5$ to investigate

the impact of β on γ in the PMS algorithm. The results in Fig. 8 show that values of γ are the same as those in the case $\beta = 4, n = 5$. From the simulation results, we can conclude that PMS can not improve the performance γ no matter what network settings are. This indicates that we have to include intruder arrival information for the sensor movement design in order to improve the performance γ .

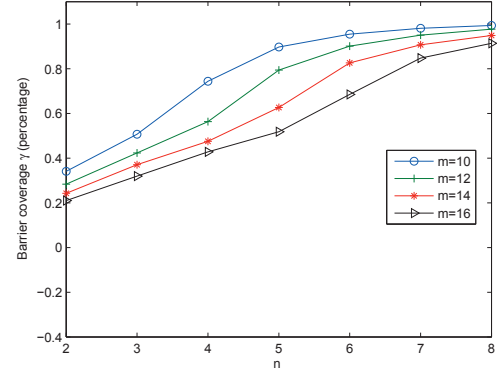


Fig. 9. Performance γ for different n and m when $\beta = 4$. For a fixed m , γ increases nonlinearly with n .

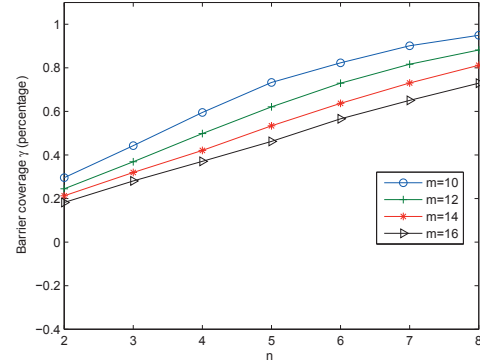


Fig. 10. Performance γ for different n and m when $\beta = 2$. More sensors are required in case $\beta = 2$ to obtain the same γ than that in the case $\beta = 4$.

We then evaluate the performance of CSP. At each slot, all the intruder arrival information obtained by each sensor will be fused together, and the available sensors will move to monitor the selected points. We first set $\beta = 4$, and calculate γ for different n and m . The results are plotted in Fig. 9. For a fixed m , γ increases nonlinearly with n . A small increase in n/m can result in a leap in γ . For example, when $n = 5, m = 10$, γ is 0.9, and when $n = 7, m = 10$, γ approaches 1. Fig. 9 also indicates γ decreases when n is fixed and m increases. Therefore, by jointly exploiting sensor mobility and intruder arrival information, performance γ can be significantly improved. Then we set β to be 2 and 6, respectively, and re-conduct the simulations to investigate the impact of β on γ . We show the results in the Figs. 10 and 11. For a fixed value of m , more sensors are required in case $\beta = 2$ to obtain the same γ than that in the case $\beta = 4$, while less sensors are needed in case $\beta = 6$. This is because β in the Weibull distribution represents the temporal correlation between two intruder arrivals. The larger the values of β , the stronger the temporal correlation. Thus less sensors are needed to obtain the same performance γ .

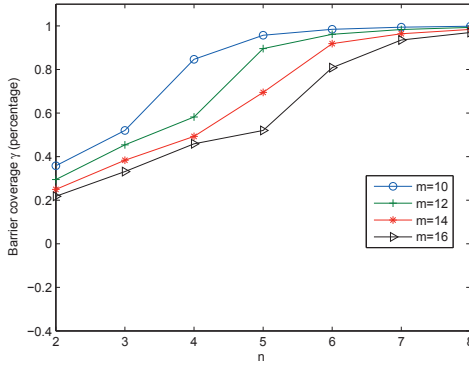


Fig. 11. Performance γ for different n and m when $\beta = 6$. Less sensors are required in case $\beta = 6$ to obtain the same γ than that in the case $\beta = 4$.

We discuss how to decide the number of mobile sensors to guarantee a predefined γ . As stated in Sec. V-C, for a predefined γ_0 , we can estimate the required number of sensors to guarantee the performance. For example, the ratios n/m in cases: i) $\beta = 2$, ii) $\beta = 4$ and iii) $\beta = 6$ should be larger than 0.6854, 0.4592, and 0.3631, respectively to ensure $\gamma \geq 0.9$. For a given m , the corresponding n can be obtained. The corresponding performances γ by setting $n/m = 0.6854, 0.4592, 0.3631$ respectively for cases i) $\beta = 2$, ii) $\beta = 4$ and iii) $\beta = 6$ are plotted in Fig. 12. We can see that when n is larger than 100, γ approaches 0.9 for all three cases. When a large scale network is involved, the calculated number of sensors can give an accurate estimate of the required number of sensors; when small number of sensors are used in the applications (e.g., $n \leq 100$), extra number of sensors can be added to guarantee the performance requirement. Therefore, the estimate of sensors provides basic information about the required number of sensors for a specific application. Fig. 12 also shows 31.46%, 54.08%, 63.69% sensors can be reduced respectively to guarantee the performance $\gamma = 0.9$ in the cases i) $\beta = 2$, ii) $\beta = 4$ and iii) $\beta = 6$. We also show the required number of sensors for cases $\beta = 1.2$, $\beta = 3$ and $\beta = 5$ in Fig. 12 and the corresponding values of n/m are 0.8592, 0.5456 and 0.4039. Note that when $\beta = 1$, the Weibull distribution reduces to Poisson distribution, according to which intruder arrivals are independent. In such case we can not improve the barrier coverage performance by leveraging the sensor mobility. Hence, we adopt $\beta = 1.2$ instead of $\beta = 1$. We conclude that by jointly leveraging sensor mobility and intruder arrival information, we can reduce a large number of sensors to achieve the same barrier coverage performance.

Finally, we study the performance γ of the two DCSP algorithms: S-DCSP and G-DCSP. In S-DCSP, after being assigned with a list of points to monitor, each sensor works independently and communicates with those in the communication range. In G-DCSP, sensors cooperate with each other in the same cluster, and we divide the sensors into 2 clusters. We set $m = 10$, and perform simulations for S-DCSP, G-DCSP and CSP under different n . The results are depicted in Fig. 13. In these three algorithms, S-DCSP has the worst performance γ , and CSP has the best. The performances γ of the three algorithms are very close, indicating the efficiency of S-DCSP and G-DCSP. The simulation results of the three algorithms in Fig. 13 for case $m = 14$ also confirm this conclusion.

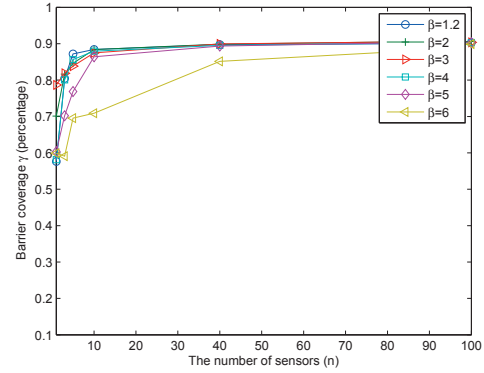


Fig. 12. An illustration of deciding the number of sensors to guarantee a predefined γ_0 .

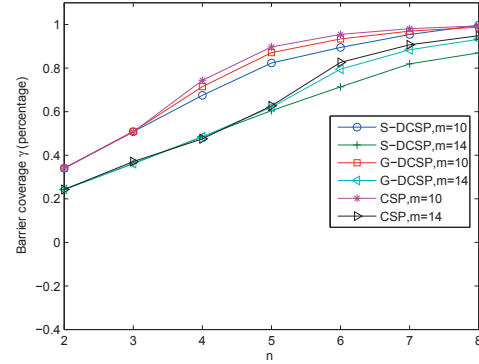


Fig. 13. Performance γ of S-DCSP, G-DCSP and CSP for $m = 10$ and $m = 14$. Performance γ of the three algorithms are very close.

B. Single Barrier case with intruder arrival following Markov chain

In this subsection, we adopt Markov chain to model the intruder arrival with the goal of showing that the proposed algorithms can also work for different intruder arrival models.

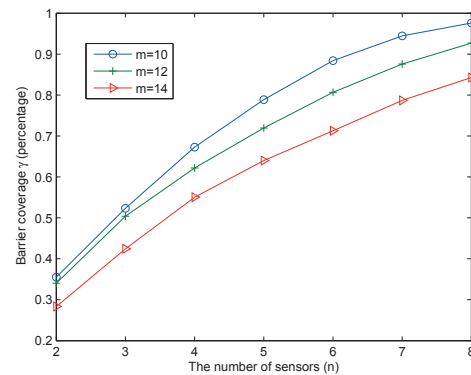


Fig. 14. Performance γ of algorithm MCSP on real data collected from spectrum usage, Vs. the number of points m and sensors n .

First, we adopt some real data collected from spectrum usage to show some practical implications of the proposed algorithms. The data record the channel states (i.e., occupancy and vacancy) of spectrum ranging from 300MHz to 3000MHz in Guangdong province, China [34]. We note that barrier coverage approach can also be applied to the scenario where we want to estimate the specific channel usage of some discrete locations: viewing channel occupancy as an intruder, we use mobile sensors (such as sensing devices piggybacked

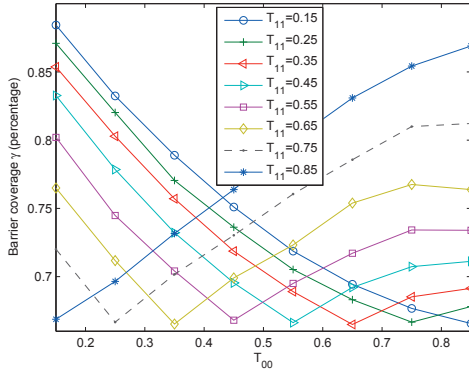


Fig. 15. Performance γ of algorithm MCSP Vs. different T_{11} and T_{00} .

on cars) to monitor specific channel states. As the resource is limited, we want to maximize the detection probability of channel occupancy by mobilizing these limited sensing devices. We adopt the Markov chain to model the real data, and Fig. 14 gives the barrier coverage performance of the modified CSP (MCSP). Again, we can see MCSP can achieve a high barrier coverage performance with a small number of sensors, which is consistent with the results obtained in the previous subsection.

To show the impact of intruder arrival patterns on the barrier coverage performance, we then proceed to perform more simulations by setting different values of the parameters in Markov chain model. Note that there are two parameters in the Markov chain model: T_{11} and T_{00} (see Eq. (9))¹. We fix $n = 8$ and $m = 12$ and vary the values of T_{11} and T_{00} from 0.15 to 0.85 with an increment of 0.1, respectively. The barrier coverage performances for all cases are plotted in Fig. 15. In Markov chain model, T_{11} and $T_{01} = 1 - T_{00}$ mean the probabilities that there is an intruder arriving in next slot if there is an intruder or no intruder arriving at the current slot, respectively. When T_{11} is close to T_{01} , then we can not know much about the intruder arrival based on the current state (as the intruder arrival probabilities are close no matter if there is an intruder arriving at current slot or not). This is confirmed by the results in Fig. 15, where a high value of $|T_{11} - T_{01}|$ yields a high barrier coverage performance and the worst performance occurs when $|T_{11} - T_{01}|$ is close to 0. For example, for curve marked as $T_{11} = 0.15$, the barrier coverage performance decreases as T_{00} increases (thus $T_{11} - T_{01}$ decreases); for curve marked as $T_{11} = 0.55$, barrier coverage first decreases and then increases because $T_{11} - T_{01}$ first decreases and then increases. Note that $\gamma = 0.667 = n/m$ when $T_{11} = T_{01} = 0.15$. This is because we do not gain any intruder arrival information and thus could not improve the barrier coverage performance. Based on these discussions, we can conclude that the proposed algorithms can effectively exploit the intruder arrival information to enhance the barrier coverage performance.

C. Two Barriers case

We perform simulations to demonstrate if the barrier coverage can be further improved by two barriers. The basic

¹The values of T_{10} and T_{01} in the transition matrix can be decided by T_{11} and T_{00} , respectively.

network settings are the same as those in the single barrier case.

Denote the number of sensors monitoring on barrier 1 by n_1 and the number of sensors on barrier 2 by n_2 . We first investigate the case when $n_1 = n_2$, i.e., two barriers are of the same importance and allocated with the same number of sensors. We plot the values of intruder detection probability under I-CSP, J-CSP and CSP when $m = 18$ in Fig. 16. I-CSP can obtain a slightly better intruder detection probability than J-CSP when n is small, and J-CSP is more desirable when n is large. However, the values of intruder detection probability under I-CSP and J-CSP are both less than those of CSP, which indicates that simply leveraging multiple barriers may not improve the barrier coverage performance. We set $m = 20$, and re-perform the simulations. The similar conclusions can be attained, which is depicted in Fig. 17.

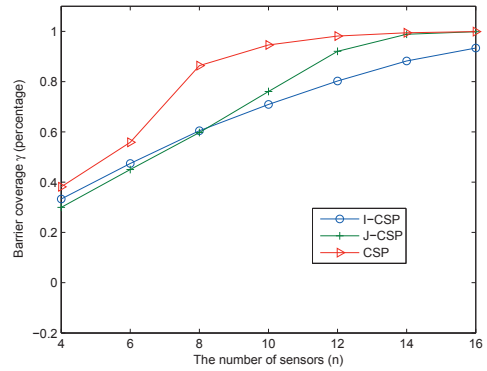


Fig. 16. The values of intruder detection probability Vs. different number of n ($n_1 = n_2 = n/2$) under I-CSP, J-CSP and CSP when $m = 18$.

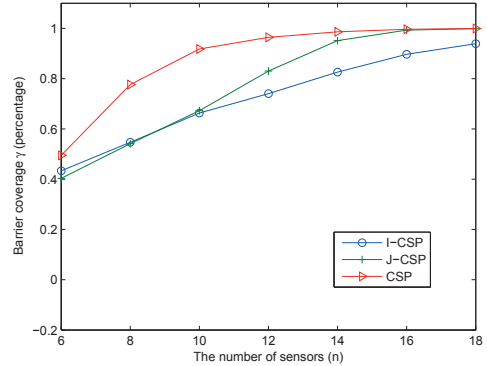


Fig. 17. The values of intruder detection probability Vs. different number of n ($n_1 = n_2 = n/2$) under I-CSP, J-CSP and CSP when $m = 20$.

We proceed to illustrate the reason why CSP outperforms J-CSP². We set the total number of points m to be 18, and the total number of sensors to be 10. We put n_1 sensors on the barrier 1 and $10 - n_1$ on the barrier 2. We vary the values of n_1 from 2 to 8, and run the simulations to obtain corresponding barrier coverage performance. The results are depicted in Fig. 18. We can see that as the number of sensors on the barrier 1 increases, the values of γ increase. Since CSP is the special case when we put all sensors on the barrier 1, this indicates CSP outperforms J-CSP in general.

²As we mentioned in Sec. VI-C, it is obvious that CSP outperforms I-CSP

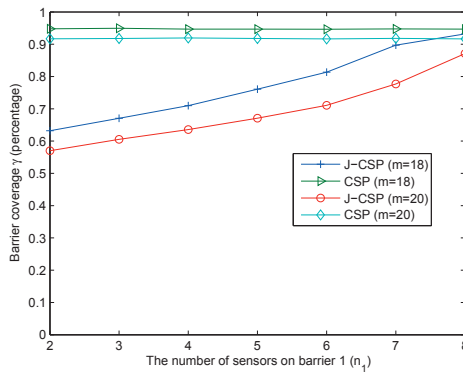


Fig. 18. The values of intruder detection probability Vs. different number of sensors n_1 on the barrier 1.

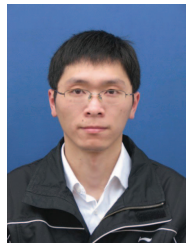
IX. CONCLUSION

We have studied the cost-effective barrier coverage problem for the case of sensor scarcity. We first designed a periodic monitoring scheduling (PMS) algorithm. Based on the insight gained from PMS, we then proposed to jointly exploit sensor mobility and intruder arrival information to improve barrier coverage. We devised a coordinated sensor patrolling (CSP) algorithm, and demonstrated that the proposed CSP can significantly enhance the barrier coverage. We also presented two distributed versions of CSP, S-DCSP and G-DCSP, to suit the decentralized nature of WSNs. We considered 2-barriers case, where sensors can move on two different barriers, and proposed two sensor movement algorithms. In addition, we generalized CSP to work for different intruder arrival models. Our simulation results indicated that the proposed algorithms can better improve the barrier coverage performance when we know more about the intruder arrival information. Our solution thus has a great potential to reduce the application budget and provides a new cost-effective approach to achieve barrier coverage in large-scale mobile sensor networks. We will study the general k -barriers coverage by mobile sensor networks in the future work.

REFERENCES

- [1] S. He, J. Chen, X. Li, X. Shen, and Y. Sun. Cost-effective barrier coverage by mobile sensor networks. In *Proceedings of IEEE Conference on Compute Communications (INFOCOM)*, 2012.
- [2] C. Huang and Y. Tseng. The coverage problem in a wireless sensor network. In *Proceedings of ACM International Workshop on Wireless Sensor Networks and Applications*, 2003.
- [3] J. Pan, L. Cai, Y. Hou, Y. Shi, and X. Shen. Optimal base-station locations in two-tiered wireless sensor networks. *IEEE Transactions on Mobile Computing*, 4(5):458–473, 2005.
- [4] E. Amaldi, A. Capone, M. Cesana, and I. Filippini. Coverage planning of wireless sensors for mobile target detection. In *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2008.
- [5] I. Dietrich and F. Dressler. On the lifetime of wireless sensor networks. *ACM Transactions on Sensor Networks*, 5(1):1–38, 2009.
- [6] J. Chen, W. Xu, S. He, Y. Sun, P. Thulasiraman, and X. Shen. Utility-based asynchronous flow control algorithm for wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 28(7):1116–1126, 2010.
- [7] R. Lu, X. Lin, H. Zhu, X. Liang, and X. Shen. Becan: A bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(1):32–43, 2012.
- [8] H. Frey and I. Stojmenovic. On delivery guarantees and worst-case forwarding bounds of elementary face routing components in ad hoc and sensor networks. *IEEE Transactions on Computers*, 59(9):1224–1238, 2010.
- [9] A. Somasundara, A. Ramamoorthy, and M. Srivastava. Mobile element scheduling with dynamic deadlines. *IEEE Transactions on Mobile Computing*, 6(4):1142–1157, 2007.
- [10] Khepera robots. Available at: <http://www.k-team.com>.
- [11] G. Wang, G. Cao, and T. Porta. Movement-assisted sensor deployment. In *IEEE Conference on Compute Communications (INFOCOM)*, 2004.
- [12] X. Li, H. Frey, N. Santoro, and I. Stojmenovic. Strictly localized sensor self-deployment for optimal focused coverage. *IEEE Transactions on Mobile Computing*, to appear, 10(11):1520–1533, 2011.
- [13] S. He, J. Chen, Y. Sun, D. Yau, and N. Yip. On optimal information capture by energy-constrained mobile sensors. *IEEE Transactions on Vehicular Technology*, 59(5):2472–2484, June 2010.
- [14] S. Kumar, T. Lai, and A. Arora. Barrier coverage with wireless sensors. In *Proceedings of International Conference on Mobile Computing and Networking (MobiCom)*, 2005.
- [15] B. Liu, O. Dousse, J. Wang, and A. Saipulla. Strong barrier coverage of wireless sensor networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2008.
- [16] G. Keung, B. Li, and Q. Zhang. The intrusion detection in mobile sensor network. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2010.
- [17] J. Li, J. Chen, and T. H. Lai. Energy-efficient intrusion detection with a barrier of probabilistic sensors. In *Proceedings of IEEE Conference on Compute Communications (INFOCOM)*, 2012.
- [18] C. Shen, W. Cheng, X. Liao, and S. Peng. Barrier coverage with mobile sensors. In *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks*, 2008.
- [19] A. Saipulla, B. Liu, G. Xing, X. Fu, and J. Wang. Barrier coverage with sensors of limited mobility. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2010.
- [20] C. Lee, D. Shin, S. W. Bae, and S. Choi. Best and worst-case coverage problems for arbitrary paths in wireless sensor networks. In *Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2010.
- [21] Y. Wang and G. Cao. Barrier coverage in camera sensor networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2011.
- [22] M. Cardei and J. Wu. Coverage in wireless sensor networks. *Handbook of Sensor Networks*, 2004.
- [23] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Journal of Wireless Ad-hoc and Sensor Networks*, 1:89–124, 2005.
- [24] N. Bisnik, A. Abouzeid, and V. Isler. Stochastic event capture using mobile sensors subject to a quality metric. *IEEE Transactions on Robotics*, 23(4):676–692, 2007.
- [25] S. Megerian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Worst and best-case coverage in sensor networks. *IEEE Transactions on Mobile Computing*, 4(1):84–92, 2005.
- [26] A. Chen, S. Kumar, and T. Lai. Designing localized algorithms for barrier coverage. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom)*, 2007.
- [27] A. Chen, T. Lai, and D. Xuan. Measuring and guaranteeing quality of barrier-coverage in wireless sensor networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2008.
- [28] A. Saipulla, C. Westphal, B. Liu, and J. Wang. Barrier coverage of line-based deployed wireless sensor networks. In *Proceedings of IEEE Conference on Compute Communications (INFOCOM)*, 2009.
- [29] G. Yang and D. Qiao. Barrier information coverage with wireless sensors. In *Proceedings of IEEE Conference on Compute Communications (INFOCOM)*, 2009.
- [30] B. Bhattacharya, M. Burmestery, Y. Hu, E. Kranakis, Q. Shi, and A. Wiese. Optimal movement of mobile sensors for barrier coverage of a planar region. *Combinatorial Optimization and Applications: Lecture Notes in Computer Science*, 5165/2008:103–115, 2008.
- [31] T. He, C. Huang, B. M. Blum, J. Stankovic, and T. Abdelzaher. Range-free localization and its impact on large scale sensor networks. *ACM Transactions on Embedded Computing Systems*, 4(4):877–905, 2005.
- [32] S. He, J. Chen, X. Li, X. Shen, and Y. Sun. Leveraging prediction to improve the coverage of wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(4):701–712, 2012.
- [33] W. Weibull. A statistical distribution function of wide applicability. *Journal of mechanics*, 18(3):293–297, 1951.

- [34] S. Yin, D. Chen, Q. Zhang, M. Liu, and S. Li. Mining spectrum usage data: a large-scale spectrum measurement study. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom)*, 2009.

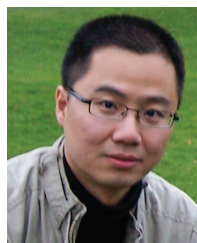


Shibo He received Ph.D degree in Control Science and Engineering from Zhejiang University in 2012. He is a member of the Group of Networked Sensing and Control (IIPC-nesc) in the State Key Laboratory of Industrial Control Technology at Zhejiang University. He was a visiting scholar at university of Waterloo from 2010 to 2011. His research interests include coverage, cross-layer optimization, and distributed algorithm design problems in wireless sensor networks.



Jiming Chen (M'08 SM'11) received B.Sc degree and Ph.D degree both in Control Science and Engineering from Zhejiang University in 2000 and 2005, respectively. He was a visiting researcher at INRIA in 2006, National University of Singapore in 2007, University of Waterloo from 2008 to 2010. Currently, he is a full professor with Department of control science and engineering, and the coordinator of group of Networked Sensing and Control in the State Key laboratory of Industrial Control Technology at Zhejiang University, China. His research interests

are estimation and control over sensor network, sensor and actuator network, target tracking in sensor networks, optimization in mobile sensor network. He has published over 50 peer-reviewed papers. He currently serves as associate editors for several international Journals. He is a guest editor of IEEE Transactions on Automatic Control, Computer Communication (Elsevier), Wireless Communication and Mobile Computer (Wiley) and Journal of Network and Computer Applications (Elsevier). He also serves as a Co-chair for Ad hoc and Sensor Network Symposium, IEEE Globecom 2011, general symposia Co-Chair of ACM IWCMC 2009 and ACM IWCMC 2010, WiCON 2010 MAC track Co-Chair, IEEE MASS 2011 Publicity Co-Chair, IEEE DCOSS 2011 Publicity Co-Chair, and TPC member for IEEE ICDCS 2010, IEEE MASS 2010, IEEE SECON 2011, IEEE INFOCOM 2011, IEEE INFOCOM 2012, etc.



Xu Li received his Ph.D. degree from Carleton University, Canada in 2008, his master degree from the University of Ottawa, Canada in 2005 and his bachelor degree from Jilin University, China in 1998, all in computer science. He held post-doc positions at the University of Waterloo, Canada, the University of Ottawa, Canada, CNRS, France and INRIA, France, where he is currently a research officer. His research interests are in wireless ad hoc, sensor and robot networks, topology control, data communications, mobility management, network security and QoS

provisioning. Dr. Li is on the editorial board of the European Transactions on Telecommunications, Ad Hoc & Sensor Wireless Networks: an International Journal, Parallel and Distributed Computing and Networks. He is a guest editor of Computer Communications (2011) and Journal of Communications (2012). He is/was in different chairing positions or among the technical program committees for many conferences and workshops, e.g., AdHoc-NOW'08-11, IEEE DCOSS'11, IEEE MASS'07&11, IEEE LCN'10&11, IEEE PIMRC'09&11, etc. He was a recipient of NSERC postdoctoral fellowship awards and a number of other awards.



Xuemin (Sherman) Shen (M'97 SM'02 F'09) received the B.Sc.(1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. Dr. Shen's research focuses on resource management in interconnected wireless/wired networks, UWB wireless communications networks, wireless network security, wireless body area networks and vehicular ad hoc and sensor networks. He is a coauthor of three books, and has published more than 400 papers and book chapters in wireless communications and networks, control and filtering. Dr. Shen served as the Technical Program Committee Chair for IEEE VTC10, the Symposia Chair for IEEE ICC10, the Tutorial Chair for IEEE ICC08, the Technical Program Committee Chair for IEEE Globecom07, the General Co-Chair for Chinacom07 and QShine06, the Founding Chair for IEEE Communications Society Technical Committee on P2P Communications and Networking. He also served as a Founding Area Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS; Editor-in-Chief for Peer-to-Peer Networking and Application; Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY; Computer Networks; and ACM/Wireless Networks, etc., and the Guest Editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premiers Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, and a Distinguished Lecturer of IEEE Communications Society.



Youxian Sun received the Diploma from the Department of Chemical Engineering, Zhejiang University, China, in 1964. He joined the Department of Chemical Engineering, Zhejiang University, in 1964. From 1984 to 1987, he was an Alexander Von Humboldt Research Fellow and Visiting Associate Professor at University of Stuttgart, Germany. He has been a full professor at Zhejiang University since 1988. In 1995, he was elevated to an Academician of the Chinese Academy of Engineering. His current research interests include modeling, control, and optimization of complex systems, and robust control design and its application. He is author/co-author of 450 journal and conference papers. He is currently the director of the Institute of Industrial Process Control and the National Engineering Research Center of Industrial Automation, Zhejiang University. He is President of the Chinese Association of Automation, also has served as Vice-Chairman of IFAC Pulp and Paper Committee and Vice-President of China Instrument and Control Society.