Improving mobile target detection on randomly deployed sensor networks

Shu Zhou

Department of Electrical and Computer Engineering,
University of New Mexico,
Albuquerque, USA
Email: francxz@ece.unm.edu

Min-You Wu

Department of Computer Science and Engineering,
Shanghai Jiao Tong University,
Shanghai, China
Email: mwu@sjtu.edu.cn

Wei Shu*

Department of Electrical and Computer Engineering,
University of New Mexico,
Albuquerque, USA
Email: shu@ece.unm.edu
*Corresponding author

Abstract: This paper investigates the detection performance of randomly deployed wireless sensor networks. It addresses the problem of detecting mobile targets that have continuous movement. We categorise these targets into two classes. The first type is rational targets, which have the knowledge of existing sensors. The other type is on the contrary. These targets neither know anything about the sensors, nor plan their path for specific purposes; therefore, their traces are straight lines. Our objective is to find a few critical positions to deploy additional sensors, so that the freedom of mobile targets can be limited. Simulation and case studies show that a few additional sensors can greatly increase the detection probability on mobile targets.

Keywords: sensor networks; mobile target; exposure; sensors; wireless networks; detection.

Reference to this paper should be made as follows: Zhou, S., Wu, M-Y. and Shu, W. (2009) 'Improving mobile target detection on randomly deployed sensor networks', *Int. J. Sensor Networks*, Vol. 6, No. 2, pp.115–128.

Biographical notes: Shu Zhou is currently working at Teledex, California, USA. He received PhD degree from the University of New Mexico, USA, in 2007. His research interests include wireless networks, sensor networks, overlay networks, multimedia networking, parallel and distributed systems, and compilers for parallel computers.

Min-You Wu is a Professor at the Department of Computer Science and Engineering at Shanghai Jiao Tong University, Shanghai, China. His research interests include wireless networks, sensor networks, overlay networks, multimedia networking, parallel and distributed systems, and compilers for parallel computers.

Wei Shu is currently an Associate Professor at the Department of Electrical and Computer Engineering, University of New Mexico, USA. Her research interests include wireless networks, sensor networks, overlay networks, multimedia networking, parallel and distributed systems, and compilers for parallel computers.

Copyright © 2009 Inderscience Enterprises Ltd.

1 Introduction

Recent advances in wireless communications and electronics have enabled development of low-cost wireless sensors. Ubiquitous networks consisting of large number of such sensors are turning into reality in the near future. It provides many new ways of gathering, processing and utilising information, and also presents new practical and theoretical challenging problems.

In the physical layer, sensor placement models studied location of sensors and its impact on the network. Most of the researches focused on power consumption, transmission and target detection. Several models (Howard et al., 2002; Wang et al., 2003; Zou and Chakrabarty, 2003a; Goldenberg et al., 2004; Zhou et al., 2004; Wang et al., 2006) were proposed in recent years, to adjust sensor distribution, in order to achieve a better network detection performance.

The earlier work on sensor network detection focused on the discrete and independent events detection. Shakkottai et al. (2002) studied coverage and connectivity, and Chakrabarty et al. (2002) studied the sensor coverage in a grid-based surveillance area. Goldenberg et al. (2004) used mobile sensors to enhance the sensor network coverage and connectivity. Dhillon et al. (2002) worked on imprecise detection coverage. Huang and Tseng (2003) and Huang et al. (2004) discussed coverage problem in both 2D and 3D scenarios. Zou and Chakrabarty (2003b) studied uncertainty-aware coverage.

Recently, researches began to expand to the area of mobile target detection. Meguerdichian et al. (2001a, 2001b) and Megerian et al. (2004) discussed maximum breath paths of mobile targets and maximum supported paths of sensors. Veltri et al. (2003) discussed the minimum and maximum exposure path. Kumar et al. (2005) discussed barrier coverage problem, preventing targets from moving in or out of the surveillance area. Then, researches expanded to a much broader scope, studying the relationship between placement and other network metrics such as power consumption (Shu et al., 2005), network lifetime (Jain and Liang, 2005), communication channel (Miorandi and Altman, 2005), bandwidth (Cheng et al., 2005), coverage ageing problem (Lee et al., 2004), etc.

Researchers also worked on how to adjust the existing sensor placement. Meguerdichian et al. (2001a) and Megerian et al. (2004) mentioned the path blocking. Liu et al. (2005) discussed the improvement on coverage by introducing mobility. Kumar et al. (2005) studied the k-barrier problem. Hou et al. (2005) studied placing relay nodes to prolong the sensor network lifetime. Toumpis and Gupta (2005) discussed the optimal sensor placement in a massively dense network.

We categorise mobile targets into two classes: *rational targets* and *free linear-moving targets*. If a target is intelligent and aware of the sensor placement, it is classified as a rational target. Such a target will design a path to reach its destination. For example, it will choose a path that is further away from dangerous spots, such as sensors, water ponds, etc., and at the same time is shortest to its destination.

The other type of targets, the free linear-moving targets, know nothing about the sensor network, nor are they able to plan their paths for any specific purposes. We assume that they move along a straight line in any direction.

There are few works specifically studying detection performance on mobile targets. In this work, we propose methods to identify critical positions that contribute the most to the freedom of mobile targets, and deploy additional sensors to limit mobile targets' activities. Specifically, for rational targets, *exposure* is used to represent the vulnerability of current sensor network. For free linear-moving targets, we adopt the concept of *Linear Uncovered Length* (*LUL*), originally proposed by Gui and Mohapatra (2004), to measure the freedom of targets. The simulation and performance data indicate that deploying only a small number of additional sensors can yield a satisfactory result in sensor network topology enhancement. This paper is organised as follows. Section 2 overviews work in Wireless Sensor Network (*WSN*) topology and mobile targets as well as explaining our motivation to apply new metrics to address the mobile targets detection problem. The mobile target detection enhancement models for two types of targets are presented in Sections 3 and 4, respectively, followed by simulation and case studies in Section 5. Finally, Section 6 concludes this paper.

2 Sensor network topology and mobile targets

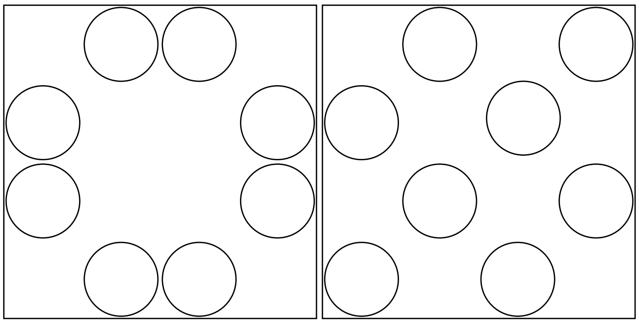
Wireless sensor network is a special type of *ad hoc* network that is tightly coupled with the environment. Sensor placement is one of the important issues that attract attentions of many scientists. It bounds the detection performance of *WSN*, and further limits other aspects of the entire network. So, changing the topology is an effective way of increasing network performances. After the early efforts of studying how to measure the performances, researchers started to think about how to improve them.

Two strategies are commonly used in sensor network topology adjustment. The first one is based on the all-mobile assumption. It assumes that all sensors are movable, so they all can reposition themselves to change the existing topology for better performance. This is an expensive approach because of the high cost of mobile sensors. Theoretically, it can lead to a globally optimised sensor distribution. In practical, it is hard to implement and achieve this optimisation. The other approach is to add a few additional sensors on top of the existing stationary ones. Generally, these additional sensors are mobile sensors (Wang et al., 2003; Zhou et al., 2004; Zhou et al., 2005). They would move to the critical positions and enhance the network topology.

Our work is based on the second partial-mobile approach, which is efficient and easy to implement. In this paper, we focus on how to identify those small number, yet critical locations that can increase the detection performance.

In the existing researches on sensor placement, *coverage* and *detection probability* are the two most commonly used metrics. Coverage is the measurement of how much of the surveillance area is under the detection of sensors. Detection probability is conceptually similar to coverage, but based on a more accurate model. However, none of them alone is sufficient to measure the mobile targets as well as to distinguish different mobile target moving patterns.

Figure 1 Limitation of coverage on free linear-moving targets detection



(a)Sensor Distribution I (b) Sensor Distribution II

The goal of this work is to enhance detection performance on mobile targets. Therefore, we need new metrics to better describe the nature of the two types of targets discussed above: *rational targets* and *free linear-moving targets*.

For the first type of targets, we can use *exposure* to measure the detection probability along their paths. Exposure is a metric especially suitable in target tracking researches. It can identify the routes with minimum detection intensity, which expose vulnerability of the network. Such paths are usually taken by the intelligent mobile targets. This motivates the rational mobile targets blocking methods in our research. To prevent mobile targets from escaping, we first determine the Most Vulnerable Paths (*MVPs*), and then select specific points along these paths as Blocking Points (*BPs*) to deploy additional sensors.

For the free linear-moving targets, there is not much work specifically addressing them. The two most commonly used metrics, coverage and detection probability, focus on the covered area, thus cannot describe the freedom of these targets. Figure 1 illustrates the limitation of coverage by comparing two different sensor distributions, where circles denote the coverage area of individual sensors. Figures 1 (a and b) both have the same coverage, while Figure 1(b) obviously has a larger hole for free linear movement. In the work of Gui and Mohapatra (2004), they proposed *Average Linear Uncovered Length* (*ALUL*) to measure the average distance a target can move before entering the detection area of some sensor. We use *ALUL* and its variations in this research to measure the freedom of mobile targets and identify the positions that obstruct their movement.

Next, we present our studies for these two types of mobile targets, *rational targets* and *free linear-moving targets*, respectively.

3 Exposure enhancement on rational targets detection

Given a randomly deployed sensor network, we consider rational targets here. A rational mobile target always attempts to find a path that is further away from sensors as much as possible. We call this path the *most vulnerable path*. The design goal of the exposure enhancement on rational targets detection is (1) to identify the *MVPs* that a mobile target would take to sneak out of the surveillance area and (2) to find the best locations to place additional sensors, subject to minimising the network vulnerability.

3.1 Sensor detection model and exposure

In this work, the Euclidean plane is used to model the surveillance area, where every position (point) on the ground is associated with a 2D coordinate. An edge between points x and y is defined as a straight line connecting them together, denoted by $e_{(x,\ y)}$. A path $p_{(a,\ b)}$ is defined as a sequence of edges connected one by one from point a to point b.

Denote the distance from sensor s_i to point x as $d(i, x)$, and the sensing ability of sensor s_i located at x as $a(i, x)$. Sensing ability $a(i, x)$ depends on the type of s_i and generally diminishes as $d(i, x)$ increases. In general, the further away a target located is from sensor s_i, the less likely it will be detected. Equation (1) is a simplified model for $a(i, x)$, where α is a manufacturer-specific parameter, and k depends on the communication scenarios, usually, $k \in (2, 4)$.

$$a(i,x) = \frac{\alpha}{d(i,x)^k}. \tag{1}$$

Next, we use *exposure* to represent vulnerability. By definition, exposure is the measurement of the sensor field intensity at a certain point or along a path. The sensor field intensity at point x is defined as below:

$$I_A(x) = \sum_{s_i \in S} a(i,x) \tag{2}$$

where S denotes the set of existing sensors.

Since the sensing ability as indicated in equation (1) drops rapidly when the distance increases, especially for a large k value, equation (2) can be further simplified by considering only the closest sensors:

$$I_c(x) = a(j,x) \tag{3}$$

where j denotes the closest sensor s_j to point x.

Based on this definition, the exposure of an edge $e_{(x,y)}$ is defined as below:

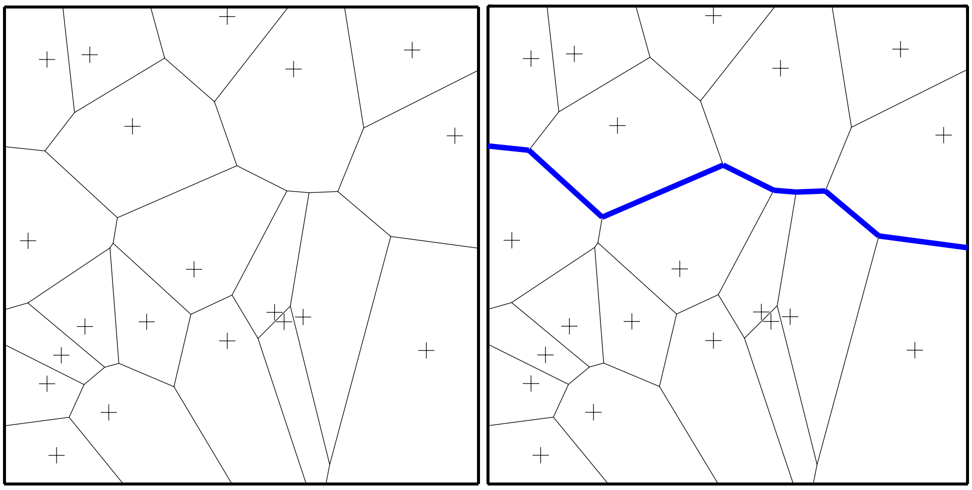
$$R_{e_{(x,y)}} = \int_x^y I_c(t)dt. \tag{4}$$

Furthermore, the exposure of a path $p_{(a,b)}$, in equation (5), is the sum of exposure of all edges along the path:

$$R_{p_{(a,b)}} = \sum_{e_{(x,y)} \in p_{(a,b)}} R_{e_{(x,y)}}. \tag{5}$$

Apparently, minimum exposure implies maximum vulnerability because of the least sensor field intensity. In the remainder of this chapter, we select the path with the minimum path exposure as the *MVP*. *Voronoi Diagram* (*VD*) can be used to find this path and calculate its vulnerability.

Figure 2 *VD* and *MVP* (see online version for colours)



(a) Voronoi Diagram (b) Most Vulnerable Path

3.2 Voronoi diagram

Voronoi diagram (Aurenhammer, 1991) is used in this research for mobile targets detecting. By *VD's* definition, for each sensor s_i in a given sensor set *S*, a polygon boundary enclosing all the intermediate points lying closer to s_i than to other sensors is called a *Voronoi Polygon* (*VP*) or *Voronoi cell*, and the set of all *VPs* is called a Voronoi diagram.

For any point on an edge of a *VD*, the distance to the nearest sensor is maximised. Figure 2(a) demonstrates this property, where plus signs and lines denote the sensors and edges of *VD*, respectively. An *MVP* is also plotted in Figure 2(b). In practice, a *VD* can be calculated at the beginning of sensor network deployment and distributed to every node. A *VD* can also be computed in a distributed manner as proposed by Bash and Desnoyers (2007) or Alsalih et al. (2008).

3.3 Paths of rational targets

The sneaking path of an intelligent target is selected to be as further from nearby sensors as possible. Thus, each edge of this path should be an edge of the *VD*. Figure 2(b) draws a rational path that horizontally cross the surveillance area, plotted by the blue/bold line segments.

An *MVP* between two vertices (or points) can be found by using Dijkstra's shortest path algorithm. In this process, a *VD* serves as the input graph, and the weight of each edge is calculated by equation (4). On the other hand, Dijkstra's algorithm cumulates each edge's weight in each step, which also satisfies equation (5). This ensures a cumulative exposure of an entire path. For an input graph $VD = (V, E)$, Dijkstra's single-source shortest path calculation with spends $O(|E|log|V|)$ time. In our research, we want to find the *MVP* across the surveillance area, i.e. between two parallel borders. So calculating shortest paths between each pair of vertices on the parallel borders could be more efficient than an all-pair Dijkstra's algorithm, which consumes $O(|V|^3)$ time.

3.4 Blocking the most vulnerable paths

Determining the *MVP* allows us to know the path that needs to be blocked. However, it certainly does not economically sounds to deploy many additional sensors along the path. We need to identify a beneficial location (or point) for a new sensor to be deployed in order to enhance the rational targets detection. We call this point the *BP*.

In order to utilise new sensors to be deployed, the *BP* should be chosen from the points along the current *MVP*. In terms of blocking, deploying a sensor on any point along *MVP* is sufficient. However, the points with extremum sensor field intensity can be of special interests. A large intensity value represents a point with a dense sensor distribution, while a small one indicates a least covered area. In these *Point*-based approaches, *BPs* are selected along the *MVP* based on their sensing intensity. Alternatively, an *Edge*-based *BP* selection is another approach. It first narrows down to a particular edge based on the exposure of the edge, then deploys a new sensor at the centre of the selected edge.

Based on either *Point*- or *Edge*-based approaches, we proposed four *BP* selection algorithms as follows:

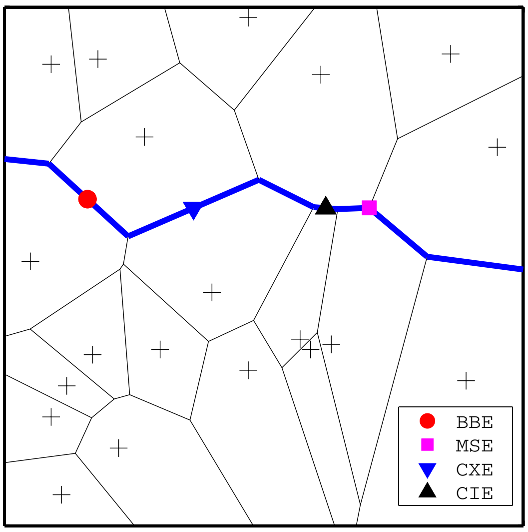
- *Best Blocking Effort* (*BBE*): An algorithm selects the position that obstructs the *MVP* the most, in other words, strengthening an area that has the greatest sensor field intensity. From all points along the *MVP*, we choose *x* with a maximum $I_c(x)$ in equation (3) as *BP*. It is an extreme approach focusing on the local maximum, with a minimum side effect on other areas.
- *Maximum Side Effect* (*MSE*): An approach considers blocking not only the current, but also the next potential *MVP*. The *BP's* location should allow the new sensor to cover an area with the least sensor field intensity. In contrast to *BBE*, *BP* is the point *x* with a minimum $I_c(x)$. This method focuses on the global performance.
- *Centre of Maximum Exposure Edge* (*CXE*): Among all edges in the *MVP*, *CXE* narrows down to edge $e_{(x,\ y)}$ with a maximum exposure $R_{e_{(x,y)}}$ in equation (4) and

then selects the centre of $e_{(x,\ y)}$ as *BP*. Similar to *BBE*, *CXE* is also an algorithm that targets thoroughly at blocking a path by obstructing the most exposed edge.

- *Centre of Minimum Exposure Edge* (*CIE*): In contrast to *CXE*, it narrows down to edge $e_{(x,\ y)}$ with a minimum exposure $R_{e_{(x,y)}}$ along the *MVP* and then selects the centre of $e_{(x,\ y)}$ as *BP*. An edge with a minimum exposure indicates a weak area in the sensor network. It can be used by multiple *MVPs*. Deploying an additional sensor there can potentially block multiple *MVPs*.

Figure 3 illustrates these four approaches, where the plus sign, solid line and bold line represent the existing sensors, the original *VD* edges and the original *MVP*, respectively. Note that in order to block a path by deploying an additional sensor at the start or the end point is usually not a good solution, since it can simply alternate the existing and all other future *MVP*s. For a rational mobile target that attempts to cross the surveillance area, it can choose another start or end point along the borders. Performances of the proposed four *BP* selection algorithms will be studied and evaluated in Section 5.

Figure 3 Illustration of four different *BP* selection algorithms (see online version for colours)



4 Blocking free linear-moving targets

Given a randomly deployed sensor network, we now consider free linear-moving targets. It is assumed that starting from its current position, a free linear-moving target will select an angle (or direction) by random, then move along a straight line, until it encounters a detection boundary of any sensor. Initially, a mobile target resides at an uncovered position, otherwise, it would be captured immediately. Thus, a contiguous area that is not within any sensor's detection range is called a *hole*. Therefore, the size and shape of a hole determines the degree of freedom for a mobile target to move without being detected. The design goal of blocking such free linear-moving targets is (1) to identify and measure the holes that existing sensors cannot cover and (2) to find the best positions to deploy additional sensors, subject to optimising the *LUL*-based metrics which will be defined in this section.

4.1 Sensor detection model and coverage

For the free linear-moving targets, we care about whether a mobile target is detectable or not. Thus, the sensor detection model of equation (1) can be simplified into a 0-1 model. That is, there are only two possibilities at every point location x: covered (or detectable) by any sensor or not. Let $c(x)$ denote the sensor coverage of sensor s_i at position x.

$$c(x) = \begin{cases} 1 & \text{if } \exists s_i,\ d(i,x) \le r_d \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where, r_d is a constant to specify the *detection range*. For simplicity, it is assumed that every sensor has the same detection range r_d. A sensor cannot detect anything that happens out of this range. This model defines a *border* of the sensor detection area, which will be used in the following definitions.

4.2 Linear uncovered length

Linear uncovered length and its derivative *ALUL* are briefly defined here, which were originally proposed by Gui and Mohapatra (2004).

- *LUL*: Let x be a location within the surveillance area and not covered by any sensor. Let $L(x,\ \theta)$ be a line segment starting at position x with an orientation angle θ. The other end point of L is at the first intersection of L with a boundary of any sensor's covering disc. The length of $L(x,\ \theta)$ defines *Linear Uncovered Length at location x with orientation θ*, denoted as $LUL(x,\ \theta)$. Figure 4 illustrates this definition.

- *ALUL* is then defined as the average length of $L(x,\ \theta)$ over all orientation angles, denoted as $ALUL(x)$.

$$ALUL(x) = \begin{cases} 0 & \text{if } c(x) \ne 0 \\ \dfrac{\int_0^{2\pi} L(x,\theta)d\theta}{2\pi} & \text{otherwise} \end{cases}. \tag{7}$$

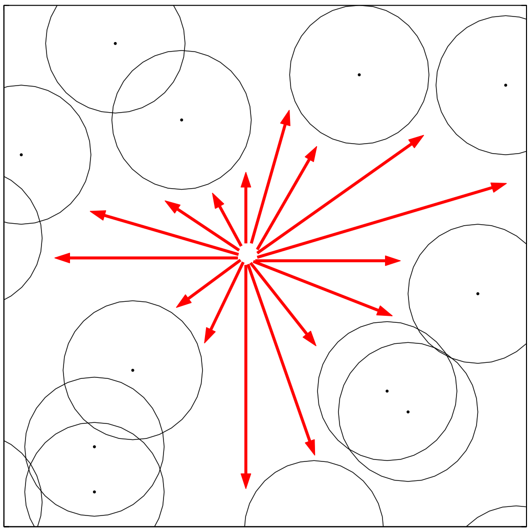
The point x^* with a maximum value of $ALUL(x)$ over all locations is the location with a largest average free-walking distance. Thus, point x^* can be an interesting point to a mobile target.

On the other hand, a surveillance sensor network needs to know how close point x to any boundary of sensor's covering discs, which is the minimum value of all $L(x,\ \theta)$. We proposed:

- *Minimum Linear Uncovered Length* (*MLUL*) is defined as the minimum value of $L(x,\ \theta)$ over all orientation angles, denoted as $MLUL(x)$.

$$MLUL(x) = \begin{cases} 0 & \text{if } c(x) \ne 0 \\ \min_{\theta\in(0,2\pi)} \{L(x,\theta)\} & \text{otherwise} \end{cases}. \tag{8}$$

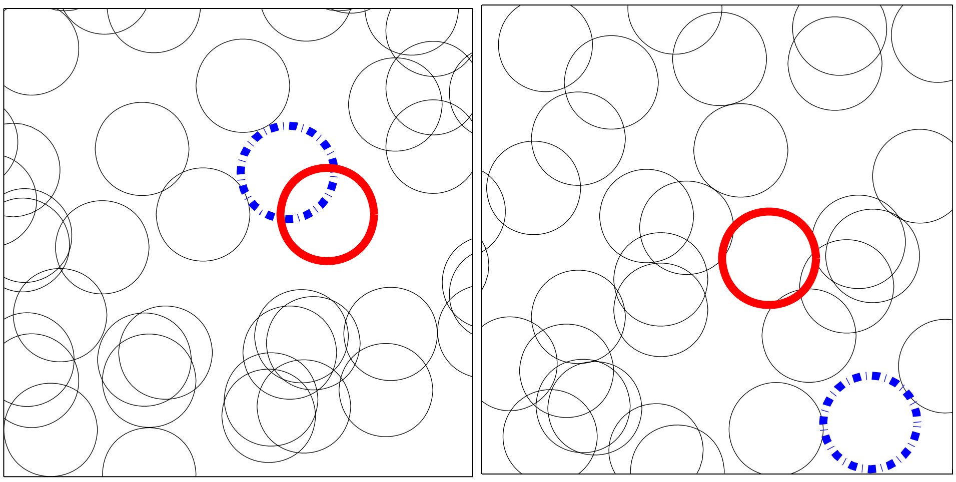
Figure 4 Illustration of *LUL* (see online version for colours)



4.3 Uncovered areas and hole patching

The goal of hole patching in a *WSN* is to determine the best positions to deploy additional sensors. Since the mobile targets move randomly, capture of the mobile targets is related to the classic coverage problem. In addition, blocking mobile targets is not only coverage-related, but also shape-related. Therefore, metrics of *LUL* needs to be considered. Figure 5 demonstrates the differences between coverage and *LUL*, where circles denote detection discs (range) of sensors. Figures 5 (a and b) show two cases with regular shaped holes and narrow shaped holes, respectively. For a coverage-based approach, many positions inside of the large uncovered area are equally good to deploy new sensors. The dotted circle represents one possible choice. However, by considering *LUL*, the centre of a large hole can be a better position, denoted by a bold circle. Similarly, the choice by considering *LUL* can be different from the one by considering the coverage only, as shown in Figure 5(b).

Figure 5 Sensor deployment: coverage-based and *LUL*-based (see online version for colours)



(a) Regular Shaped Hole (b) Narrow Shaped Hole

4.3.1 The centralised position selection

In order to determine the best position for insertion of a new sensor, the *ALUL*(x) or *MLUL*(x) of every point in the surveillance area needs to be calculated first. Then, a new sensor should be deployed at the position with maximum *ALUL*(x) or *MLUL*(x). The process is repeated for deployment of additional sensors. Since the procedure cannot be applied continuously for an uncovered area, the surveillance area is required to be divided into grids.

Figure 6 lists the basic steps of the *ALUL*/*MLUL* calculation and selection procedure. The computation complexity is $O(\|A\|^2 N)$, where $\|A\|$ denotes the total number of grids in area A, and θ is increased by user customised steps. This algorithm has been evaluated in Section 5 with a satisfying performance. Here, for *MLUL*, the computation of *MLUL*(x) on any point x is limited by the sensors closest to x, so only local information is required. However, the overall algorithm is centralised. Considering the issue of scalability, this algorithm needs further elaboration into a distributed algorithm.

Figure 6 Calculating the location for the next new sensor

Location-Selection
  For every uncovered point (grid) x on area A
  init $D[0, 2\pi] = 0$
    For every direction $\theta \in (0, 2\pi)$
      calculate $L(x, \theta)$
      record the length of $L(x, \theta)$ in $D[\theta]$
    compute $ALUL(x)$ or $MLUL(x)$
  Select the x with maximum $ALUL(x)$ or $MLUL(x)$

4.3.2 The distributed position selection

As a distributed algorithm, every sensor only calculates the *MLUL*(x) of nearby area. We use a *VD* (Aurenhammer, 1991) to divide the whole surveillance area into small *VP* regions (larger than grids). Each region contains one sensor, s_i, which is responsible for computing the *MLUL*(x) of points (grids) within that *VP* region v_i.

Since sensor s_i is the one closest to every point inside of a corresponding *VP* region, this distributed algorithm naturally distributes the workload into different sensors. It greatly reduces complexity and allows our method to scale when the number of sensors grows large. However, each *VP* region is still required to divided into grids for calculation. When the size of the surveillance area grows but not the number of sensors, the computation amount for each sensor can be substantially increased too.

This problem can also be relaxed by utilising characteristics of *VD*. Note that the vertices of a *VP* region v_i exhibit some special properties: they are further away from sensor s_i compared to the inside ones, therefore, will be points with maximum $MLUL(x)$ among all points in that *VP* region, as approved by the following theorem.

Lemma 4.1: *The vertex of a VP that is furthest away from the owner sensor has the maximum MLUL(x) among all points in this VP.*

Proof: By the definition of *VP* and *VD*, for any point x inside of a *VP* v_i, the closest sensor is the corresponding owner sensor s_i. So, the $MLUL(x)$ is limited and only limited by s_i.

$$MLUL(x) = \begin{cases} 0 & \text{if } c(x) \neq 0 \\ \min_{\theta \in (0,2\pi)} \{L(x,\theta)\} & \text{otherwise} \\ \quad = d(i,x) - r_d & \end{cases} \tag{9}$$

where, r_d is the detection range. As the boundary of a *VP*, an edge is the set of points that are furthest away from s_i. So, the vertex x_m with maximum $d(i, x_m)$ can yield the maximum $MLUL(x)$:

$$\max\{d(i,x)\} \equiv \max\{MLUL(x)\}, (x \in v_i) \tag{10}$$

Lemma 4.2: *The vertex x_m selected in Lemma 4.1 is the best position to deploy a new sensor to reduce the maximum MLUL(x) of VP region v_i.*

Proof: Suppose we add the new sensor at a point x' other than x_m. Then:

Case 1: the detection range of the new sensor does not overlap the $MLUL(x_m)$ circle. By definition, the $MLUL(x_m)$ in v_i is kept the same.

Case 2: the detection range of the new sensor overlaps the $MLUL(x_m)$ circle. Then there is a blank area on some far side of sensor s_j. This area has a larger new maximum $MLUL(x)$ than that of a sensor on x_m because $d(j, x_m) < d(j, x')$.

Combining these two lemmas, hereby we present the theorem as follows.

Theorem 4.3: *The vertex in a VD that is furthest away from all sensors is an optimal position to deploy a new sensor to reduce the maximum MLUL(x) over the whole surveillance area.*

Based on the above theorem, Figure 7 briefly describes a distributed *MLUL* selection algorithm. As discussed previously, a *VD* can be calculated at the beginning of sensor network deployment and distributed to every node. Or, a *VD* can also be computed in a distributed manner as proposed by Bash and Desnoyers (2007) or Alsalih et al. (2008). Generating *VD* requires $O(NlogN)$ time, plus the sorting time $O(NlogN)$; it is a $O(NlogN)$ algorithm. The other advantage of this algorithm is that, it is not based on a grid-based model, so the result is precise. Note that each edge of a *VP* is shared by two sensors, and each vertex is shared by more than two sensors. A collision-avoidance mechanism is required to solve this problem. For example, as all sensors know their (relative) locations, they can use their coordination to avoid collision. Other generic election algorithms may also be applied.

Figure 7 Distributed maximum $MLUL(x)$ selection algorithm

```
Distributed-MLUL-Selection
    Generate VD
    For every vertex x in VD
        MLUL(x) = d(i,x) - r_d
    sort vertices of VD by MLUL(x)
    Select the x with maximum MLUL(x)
```

5 Performance studies

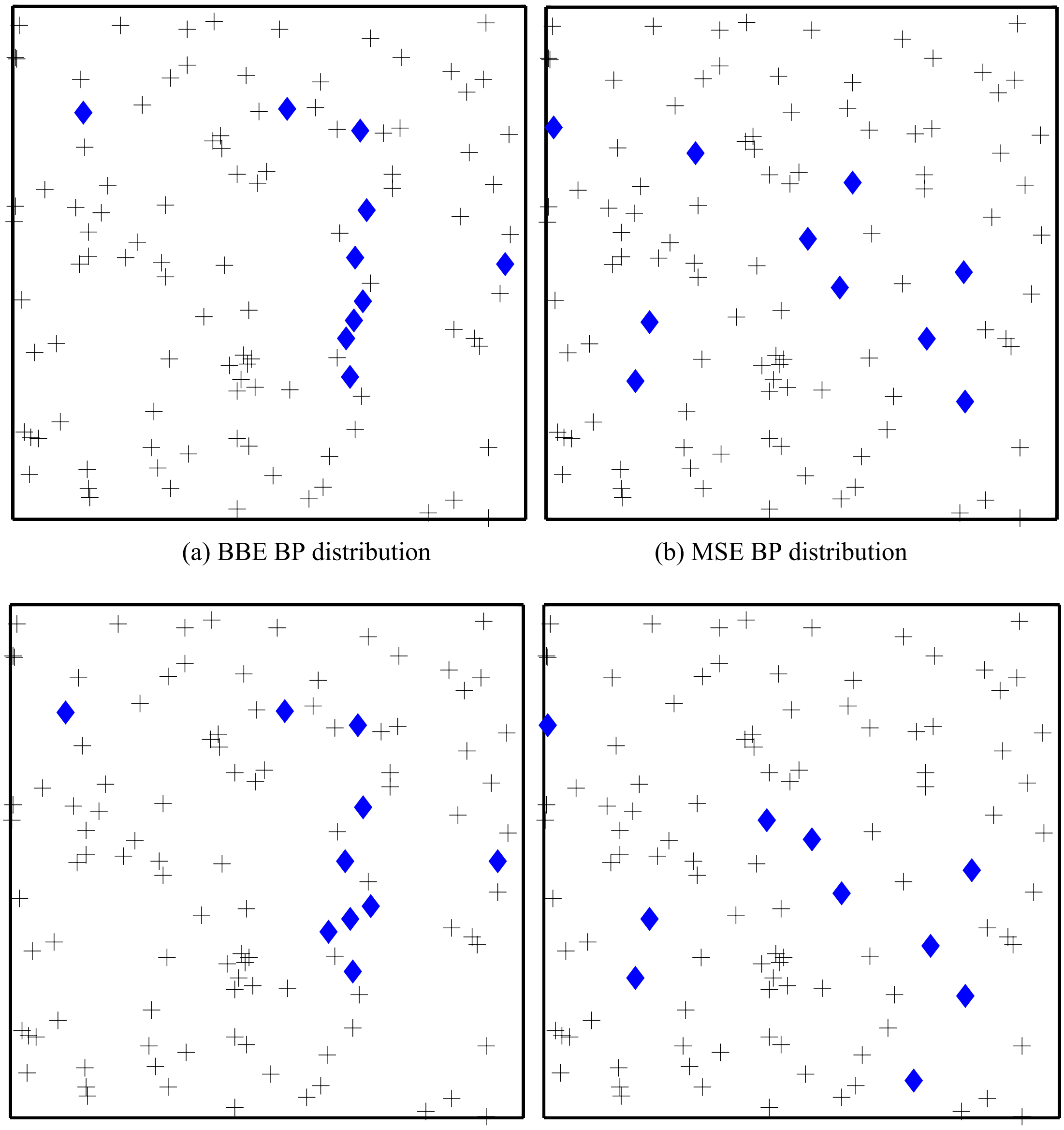
We have conducted simulations on *WSNs* to study performance of our work. The testbed consists of 100 stationary sensors that are randomly deployed into a square surveillance area (T = 400 × 400), following a uniform distribution. For the sensing model, we selected k = 2, α = 1 and r_d = 20. Simulations are performed for different types of mobile targets, respectively.

5.1 Exposure-based blocking of rational mobile targets

For rational mobile targets, we designed two test cases: *Escaping* and *Crossing*. In the escaping case, a mobile target attempts to move out of the surveillance area in any chosen direction from its initial position near by the centre. In the crossing case, a mobile target makes an effort to move across the surveillance area. We do not know whether it is moving either horizontally or vertically. Thus, we calculate two *MVPs*, one for each direction.

Figure 8 shows a sample sensor distribution in escaping case after ten new sensors have been newly deployed. The plus signs and the large solid diamonds represent the original sensors and the newly added sensors, respectively. Note that, *BBE* adds most new sensors on one direction of the escaping path, and blocks that path thoroughly. At the same time, it also consumes many sensors on one direction and has not left enough sensors to take care of other directions. *CXE* is similar to *BBE*. *MSE* behaves differently compared to the *BBE* or *CXE*, and performs better in terms of sensor utilisation. Here, the new sensors are placed in sparse areas. There is hardly any easy crossing path left. *CIE* performs closely to *MSE*.

Figure 8 *BP* distribution in escaping scenario (see online version for colours)



(a) BBE BP distribution (b) MSE BP distribution

(c) CXE BP distribution (d) CIE BP distribution

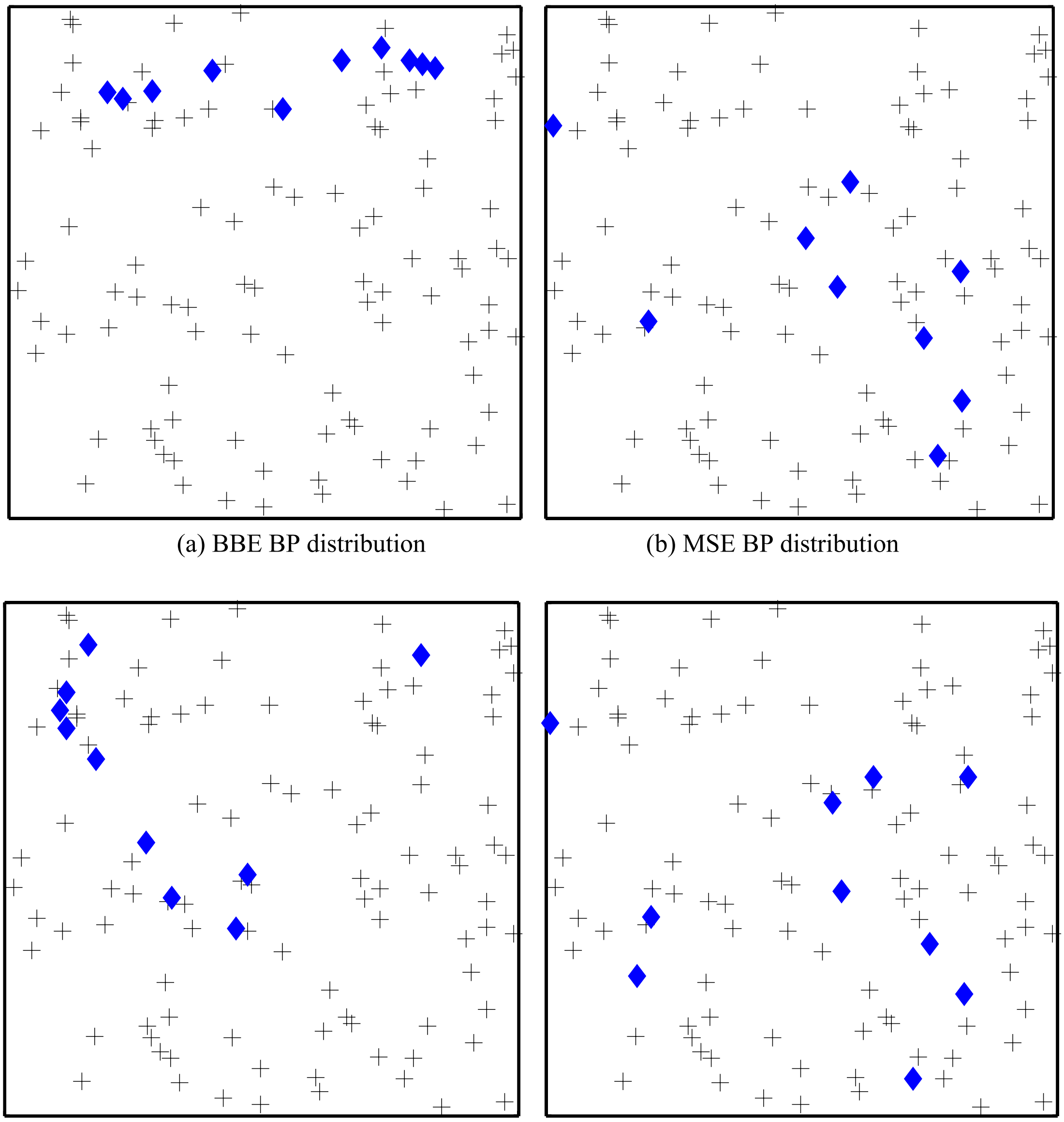
Figure 9 is a sample of sensor distribution in the crossing case after ten new sensors have been newly deployed. The plus signs and the large solid diamonds represent the original sensors and the newly added sensors, respectively. The result is similar to that of the escaping case.

For more accurate numerical results, we generated ten randomly deployed sensor networks, using the same setting as described in the beginning of this section. All the following test results in this subsection were averaged over these ten test cases.

Figure 10 plots the performances of our algorithms in the escaping case. It further confirmed the result of Figure 8. Four *BP* selection algorithms are divided notably into two groups. *MSE* and *CIE* outperform *BBE* and *CXE*, especially when the number of additional sensors increases.

Figure 11 plots the performances in the crossing case. No matter the average or the minimum exposure of horizontal and vertical *MVP*, *MSE* and *CIE* again performed better than the other two. Same as in Figure 10, *MSE* performs best. Together with Figure 9, the results of the crossing case are consistent with that of the escaping case. We can also conclude that, *minimum extremum* based approaches outperform *maximum extremum* based approaches.

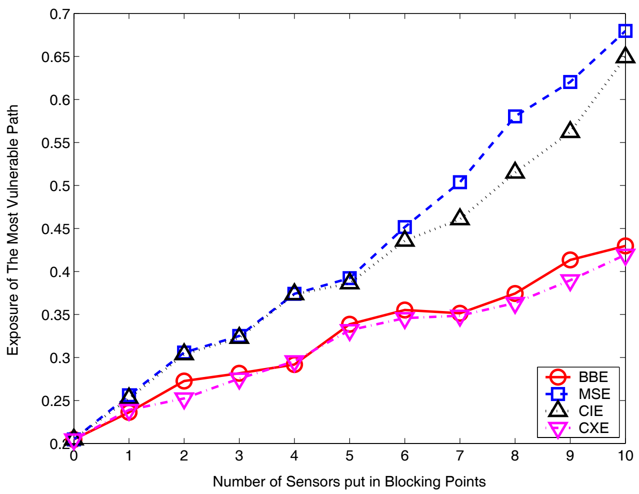
Figure 9 *BP* distribution in crossing scenario (see online version for colours)



(a) BBE BP distribution (b) MSE BP distribution

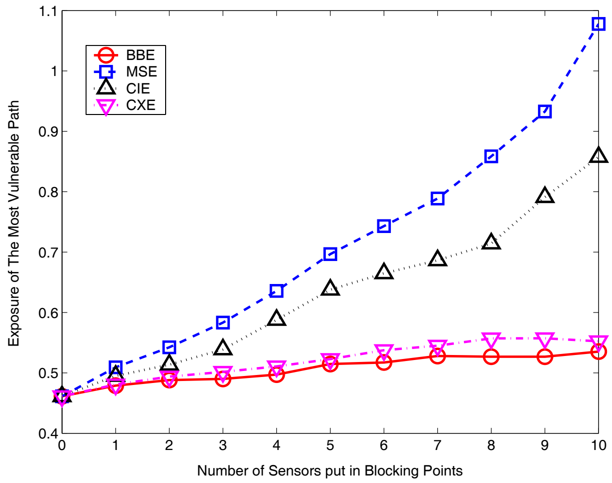
(c) CXE BP distribution (d) CIE BP distribution

Figure 10 Detection performance on rational targets in escaping scenario (see online version for colours)

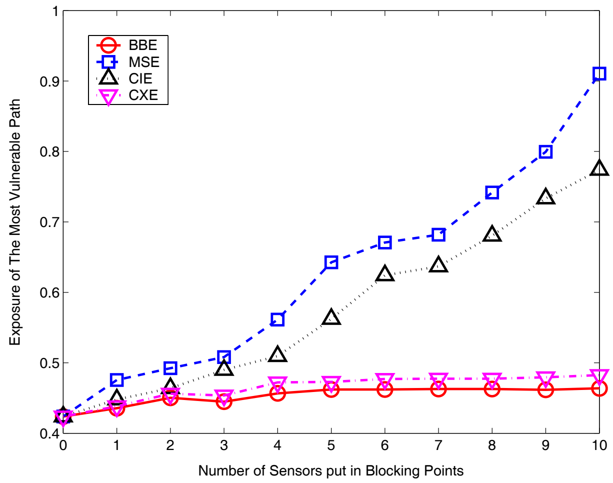


The major difference between this work and the traditional coverage problem is that the targets are moving rationally and know how to avoid existing sensors. For comparison, we present simulation results of coverage-based sensor deployment in this rational movement scenario, by considering both escaping and crossing cases. The results are also averaged over the same ten test cases. Figure 12 compares the performance of coverage-based approach with our *BP*-selection algorithms. Evidently, the coverage-based approach cannot handle a mobile target's rational movement properly. In escaping case, rational mobile targets can escape in any direction. The coverage-based sensor deployment can hardly block all potential sneaking paths. Therefore, the curve of the coverage-based algorithm in Figure 12(a) lies at the bottom. We have considered both horizontal and vertical paths in the crossing case. Under this condition, the coverage-based approach performs closely to *BBE* and *CXE*, as shown in Figures 12 (b and c). However, it is still not comparable to *MSE* and *CIE*.

Figure 11 Detection performances on rational targets in crossing scenario (see online version for colours)

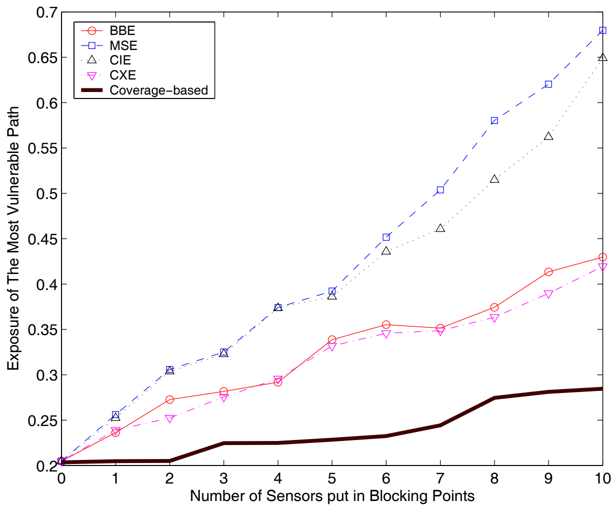


(a) Average Exposure of Horizontal and Vertical MVP

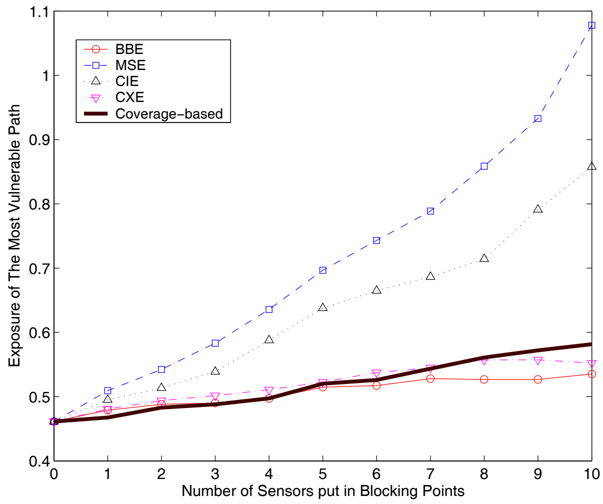


(b) Minimum Exposure of Horizontal and Vertical MVP

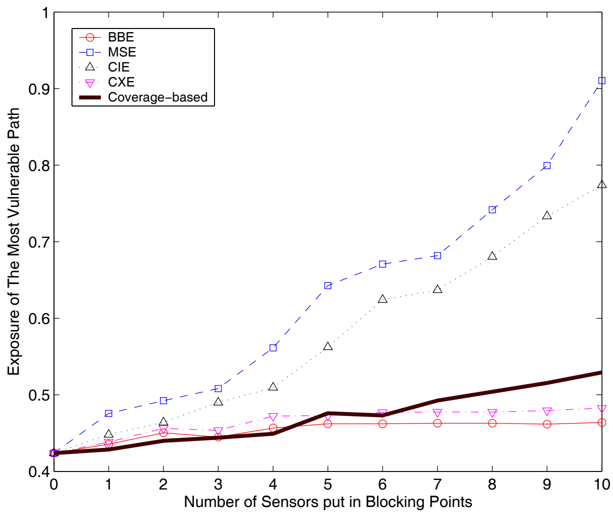
Figure 12 Performance comparison: exposure- and coverage-based approaches (see online version for colours)



(a) Exposure of MVP (Escaping)



(b) Average Exposure of Horizontal and Vertical MVP (Crossing)



(c) Minimum Exposure of Horizontal and Vertical MVP (Crossing)

In summary, a proper vulnerability blocking approach can perform well in both escaping and crossing scenarios. In our simulations, by adding only ten extra sensors into an existing sensor network with 100 sensors, *MSE* yields more than 70% exposure increase in each test case. The exposure of *MVP* increases from 0.14 to 0.37 in the escaping case. The average exposure of horizontal and vertical *MVPs* increases from 0.37 to 0.64, while their minimum exposure increases from 0.37 to 0.63. We also found out that the algorithms of selecting minimum exposure locations as *BP* would be better than those of selecting maximum exposures. *BBE* intended to deploy sensors on some already jammed edges but to ignore other areas, resulting in with weak performance. *MSE* and *CIE* not only blocked the current *MVP*, but also covered the most uncovered area to block other potential *MVPs*. They would not be trapped by a local minimum and therefore, have shown a good performance in both scenarios. Comparison to a coverage-based approach further demonstrated advantages of path-based approaches to detecting rational movement of mobile targets.

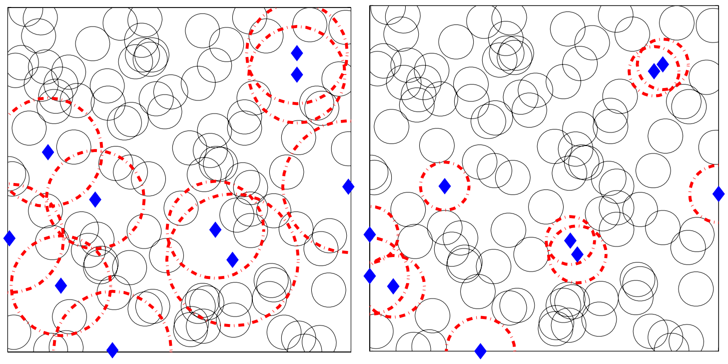
5.2 LUL-based blocking of free linear-moving targets

The simulation on blocking free linear-moving targets is based on the same setting as the previous simulation. Figure 13(a) is an example of how an *ALUL*-based hole-patching algorithm performs, where solid diamonds denote some points with large *ALUL*(x) and the radius of bold dashed circles represents the value of *ALUL*(x). As we can see, the selected points are all in the wide, uncovered holes, although not being at the centres of those holes. Some special points effectively cut off places that allow long, straight free moving lines. Figure 13(b) demonstrated the performance of *MLUL*-based hole-patching algorithm. Due to the differences between *ALUL* and *MLUL*, the selected points are closer to centres of big holes.

Once again, we generated ten sets of randomly deployed sensors, using the same setting as described at the beginning of this section. The following tests and data were averaged over these test sets.

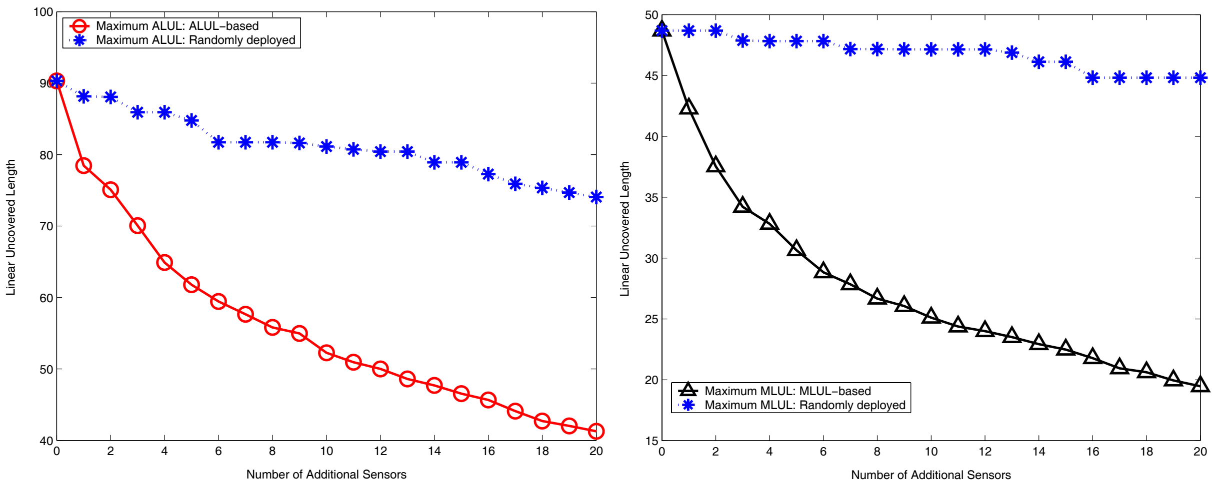
Figure 14(a) illustrated the performance of *ALUL*-based hole-patching algorithm. The curve shows a substantial drop of the maximum *ALUL* with increasing number of additional sensors. The averaged maximum *ALUL*(x) value drops from 90.31 to 41.27, which is more than 54%. As a reference, we also plot the performance of random deployment of up to 20 sensors. The random deployment can hardly hit the right position, so the averaged maximum *ALUL*(x) value reduced slowly. The big gap between these two curves further confirmed our hole-patching scheme. Similar to Figure 14(a), 14(b) demonstrated good performance of *MLUL*-based hole-patching algorithm. The averaged maximum value of *MLUL*(x) drops from 48.69 to 19.47, which is about 60%. Again, its performance advantage over the random sensors deployment is remarkable.

Figure 13 Comparison of *ALUL*- and *MLUL*-based hole patching (see online version for colours)



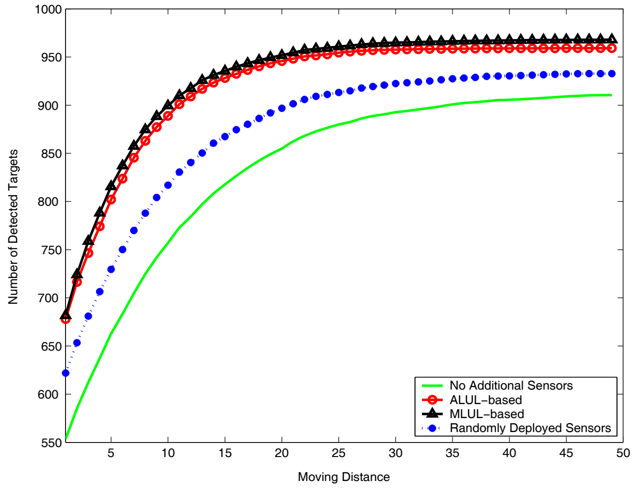
(a) ALUL-based (b) MLUL-based

Figure 14 Performance of *ALUL*- and *MLUL*-based hole patching (see online version for colours)



(a) ALUL-based (b) MLUL-based

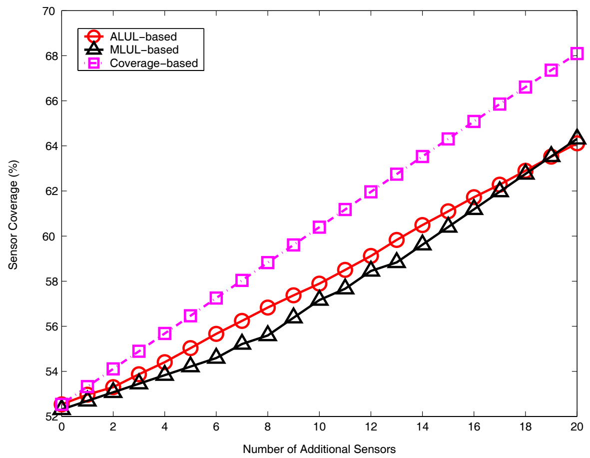
Figure 15 Random walk test on *ALUL* and *MLUL* (see online version for colours)



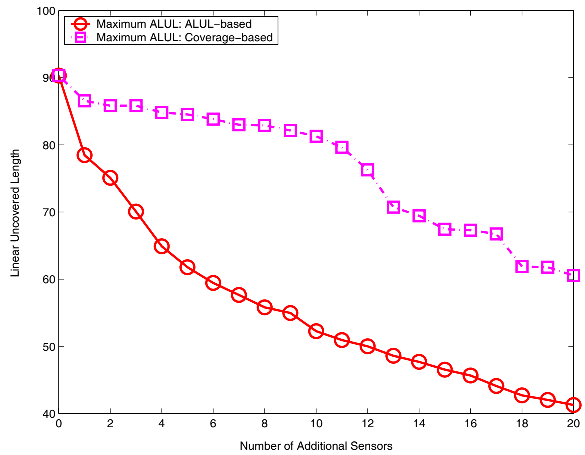
In order to compare *ALUL*- and *MLUL*-based algorithms, a random walk test is designed: 1000 mobile targets move from 1000 random positions, with randomly chosen angles. In Figure 15, we plot the number of detected targets along with the increase in the moving distance. Same as before, the data were averaged over ten sets of randomly deployed sensors. Both algorithms show a significant improvement in their detection capability, especially when the mobile targets have not moved too far away. *MLUL* slightly outperformed *ALUL*, which validated our goal of using *MLUL*.

In addition to testing *ALUL*- and *MLUL*-based approaches themselves, we also compared their performance with the classic coverage-based heuristics. As discussed in the previous sections, blocking of the free linear-moving targets is not only coverage-related, but also shape-related. We compared the performance of our proposed methods with a coverage-based approach. A sample of their different deployment methods is demonstrated in Figure 5. The simulation results of the traditional sensor coverage, the averaged maximum *ALUL* and *MLUL* are plotted in Figures 16 (a–c), respectively. The data also were averaged over ten sets of randomly deployed sensors. Figure 16(a) shows that both *ALUL*- and *MLUL*-based algorithms have good performance on sensor coverage. When compared with a coverage-based approach, their performance is close, only about 4% less than the optimal result. At the mean time, they both remarkably outperform the coverage-based algorithm in blocking big holes, as shown in Figures 16 (b and c). The wide gaps between coverage and *ALUL*/*MLUL* curves indicated that the coverage-based approach cannot effectively identify the big holes (the most needed places) in the area.

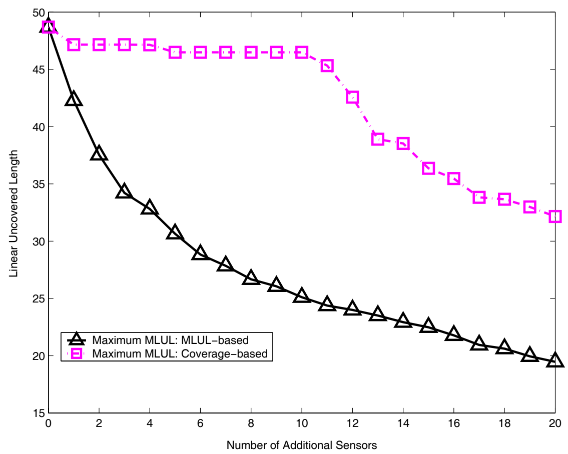
Figure 16 Performance comparison: *LUL*- and coverage-based approaches (see online version for colours)



(a) Sensor Coverage



(b) ALUL



(c) MLUL

The performance of our hole-patching mechanism is satisfactory. Both *ALUL*- and *MLUL*-based algorithms can significantly restrict the free linear movement of mobile targets. Moreover, the *MLUL*-based algorithm is less computational expensive, more accurate, easier to implement and also outperforms the *ALUL*-based one. The performance of randomly deployed sensors is not comparable to those of *ALUL* or *MLUL*.

6 Conclusion and future works

This paper presents two new methods of increasing sensor network detection for mobile targets. One concentrates on decreasing sensor network vulnerability, the other targets at the undetected area of free linear movements.

For rational mobile targets, we use exposure as the metric to represent network vulnerability, and identify the corresponding paths. We select a few points from the *MVPs* and deploy additional sensors there to block these paths. Several *BP* selection algorithms are studied. Simulation results in both escaping and crossing cases demonstrate a significant exposure increase. It is the first work that concentrates on detection of path-based rational mobile targets and sensor network vulnerability.

For free linear-moving targets, we propose a new metrics, *minimum linear uncovered length*, to detect and measure big holes of the sensor network. It allows a fast computing, accurate results, good performance and easy to be localised. The simulation and test cases demonstrate good performance of this method on hole patching, mobile targets restricting and sensor coverage.

This work points out some open questions in the mobile targets detection problem. The *random walk* moving pattern could be another good model for those targets. Other interesting topics may include the habit of mobile targets, the holes due to the limitations of sensor's communication range, the heterogeneity of sensing model, etc.

Acknowledgements

This work was supported in part by the Natural Science Foundation of China under Grant 60573138 and Grant 60773091, by the National Grand Fundamental Research 973 Program of China under Grant 2006CB303000, by the National 863 Program under Grant 2006AA01Z247 and by the United States National Science Foundation under Grant CNR-0626380.

References

Alsalih, W., Islam, K., Núñez-Rodríguez, Y. and Xiao, H.Y. (2008) 'Distributed Voronoi diagram computation in wireless sensor networks', *Proceedings of the 20th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

Aurenhammer, F. (1991) 'Voronoi diagrams – a survey of a fundamental geometric data structure', *ACM Computing Surveys*, Vol. 23, No. 3, pp.345–405.

Bash, B.A. and Desnoyers, P.J. (2007) 'Exact distributed Voronoi cell computation in sensor networks', *International Conference on Information Processing in Sensor Networks (IPSN '07)*, Cambridge, MA.

Chakrabarty, K., Iyengar, S., Qi, H. and Cho, E. (2002) 'Grid coverage for surveillance and target location in distributed sensor networks', *IEEE Trans on Computers*, Vol. 51, No. 12, pp.1448–1453.

Cheng, M., Ruan, L. and Wu, W. (2005) 'Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks', *IEEE Infocom 2005*.

Dhillon, S.S., Chakrabarty, K. and Iyengar, S.S. (2002) 'Sensor placement for grid coverage under imprecise detections', *International Conference on Information Fusion (FUSION 2002)*, pp.1581–1587.

Goldenberg, D.K., Lin, J. and Morse, A.S. (2004) 'Towards mobility as a network control primitive', *The Fifth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '04)*, pp.163–174.

Gui, C. and Mohapatra, P. (2004) 'Power conservation and quality of surveillance in target tracking sensor networks', *ACM Mobile Computing and Networking*, pp.129–143.

Hou, Y.T., Shi, Y., Sherali, H.D. and Midkiff, S.F. (2005) 'Prolonging sensor network lifetime with energy provisioning and relay node placement', *IEEE SEC ON 2005*.

Howard, A., Mataric, M.J. and Sukhatme, G.S. (2002) 'Mobile sensor network deployment using potential fields: a distributed scalable solution to the area coverage problem', *The 6th International Conference on Distributed Autonomous Robotic Systems (DARS 02)*, pp.299–308.

Huang, C. and Tseng, Y. (2003) 'The coverage problem in a wireless sensor network', *ACM Wireless Sensor Networks and Applications Workshop*, pp.115–121.

Huang, C., Tseng, Y. and Lo, L. (2004) 'The coverage problem in three-dimensional wireless sensor networks', *IEEE GlobeCom 2004*, pp.3182–3186.

Jain, E. and Liang, Q. (2005) 'Sensor placement and lifetime of wireless sensor networks: theory and performance analysis', *IEEE Globecom 2005*.

Kumar, S., Lai, T. and Arora, A. (2005) 'Barrier coverage with wireless sensors', *ACM MobiCom 2005*.

Lee, J., Krishnamachari, B. and Kuo, C.J. (2004) 'Impact of heterogeneous deployment on lifetime sensing coverage in sensor networks', *IEEE SECON 2004*.

Liu, B., Brass, P., Dousse, O., Nain, P. and Towsley, D. (2005) 'Mobility improves coverage of sensor networks', *ACM MobiHoc 2005*.

Megerian, S., Koushanfar, F., Potkonjak, M. and Srivastava, M. (2004) 'Worst and best-case coverage in sensor networks', *IEEE Transactions on Mobile Computing*, Vol. 4, No. 1, pp.84–91.

Meguerdichian, S., Koushanfar, F., Potkonjak, M. and Srivastava, M. (2001a) 'Coverage problems in wireless ad-hoc sensor networks', *IEEE InfoCom 2001*.

Meguerdichian, S., Koushanfar, F., Qu, G. and Potkonjak, M. (2001b) 'Exposure in wireless ad-hoc sensor networks', *ACM Mobile Computing and Networking*.

Miorandi, D. and Altman, E. (2005) 'Coverage and connectivity of ad hoc networks in presence of channel randomness', *IEEE InfoCom 2005*.

Shakkottai, S., Srikant, R. and Shroff, N.B. (2002) 'Unreliable sensor grids: coverage, connectivity and diameter', *CSL Technical Report.*

Shu, T., Krunz, M. and Vrudhula, S. (2005) 'Power balanced coverage-time optimization for clustered wireless sensor networks', *ACM MobiHoc 2005*.

Toumpis, S. and Gupta, G. (2005) 'Optimal placement of nodes in large sensor networks under a general physical layer model', *IEEE SECON 2005*.

Veltri, G., Huang, Q., Qu, G. and Potkonjak, M. (2003) 'Minimal and maximal exposure path algorithms for wireless embedded sensor networks', *1st ACM International Conference on Embedded Networked Sensor Systems*.

Wang, G., Cao, G. and Porta, T.L. (2003) 'A bidding protocol for deploying mobile sensors', *11th IEEE International Conference on Network Protocols*.

Wang, G., Cao, G. and Porta, T.L. (2006) 'Movement-assisted sensor deployment', *IEEE InfoCom 2006*.

Zhou, S., Wu, M-Y. and Shu, W. (2004) 'Finding optimal placements for mobile sensors: wireless sensor network topology adjustment', *IEEE 6th CAS Symposium on Emerging Technologies*.

Zhou, S., Wu, M-Y. and Shu, W. (2005) 'Terrain-constrained mobile sensor networks', *IEEE Globecom 2005*.

Zou, Y. and Chakrabarty, K. (2003a) 'Sensor deployment and target localization based on virtual forces', *IEEE InfoCom 2003*.

Zou, Y. and Chakrabarty, K. (2003b) 'Uncertainty-aware sensor deployment algorithms for surveillance applications', *IEEE GlobeCom 2003*.