An Efficient metric to evaluate the coverage quality of multiple-view barrier in wireless camera sensor networks

Abstract

Barrier coverage problems in Wireless Camera Sensor Network (WCSN) have attracted increasing interest by academic community because of their huge potential in applications. Almost traditional barrier coverage problems focus on detecting object crossing the barrier sensors. However, only detecting the intruder may be not sufficient, i.e. in terrorists recognized when detailed information about the object such as image at a certain view is required. This study aims at analyzing conditions to achieve barrier coverage under a novel multiple view coverage model in WCSN (hereinafter MVBC problem). The MVBC problem ensures both detecting intruder and recognizing the face of intruder when the intruder comes cross the sensor barrier while requires only a fixed number of sensor nodes. The dynamic partition method is then proposed to solve the MVBC problem. By conducting intensive experiments, the result indicate that our proposed method provides high quality and stable solutions. We then devise a measure as differentiation coverage function to estimate quality of the sensor barrier obtained. Theoretically we show that our differentiation field intensity function is more practical than other existing intensity functions. The empirical results also show that our new measure provides a good metrics for estimating how well information of an object crossing the barrier can be obtained.

Keywords: Multiple-view coverage model, multiple-view barrier coverage, wireless camera sensor networks, differential field intensity

1. Introduction

Recently, with multimedia technique development, wireless camera sensor networks (WCSN) have drawn the attention of research community [1-6]. WCSNs can harvest much richer information of the environment in the forms of images or videos than conventional scalar sensors, and thus promise an extremely potential in applications, e.g. boundary surveillance (national border, critical resource protection) and intrusion detection. Especially, security surveillance or intrusion detection application, the barrier coverage problem in WCSNs is expected to build up camera sensor barriers efficiently such that every image of intruder can be gathered more details. However, this problem is much different and more complicated than the conventional barrier coverage problem because of unique features of WCSNs such as limited sensing angle, directional sensing, communicating range and line of sight. When sensing range of a chain of camera sensors across the surveillance region is simply combining, that does not provide effective barrier coverage. Because an intruder may cross the barrier without being recognized, i.e. its face image could not be caught. However, most prior researches about the barrier coverage problems aim at building as many as sensor barriers with efficient cost to only detect intruder. In reality, only detecting object is not enough in some scenarios, i.e. there was a terrorist at the airport; an unauthorized intruder penetrates the monitored region. In such circumstances, a face recognition system is rather needed. To overcome this drawback, this paper investigates a novel barrier coverage problem in WCSNs, which verifies whether any given WCSNs can be formed a sensor barrier such that any intruder goes cross the barrier of sensor to be both detecting and recognizing.

To monitor stable objects with ensuring that the face of object is always caught by at least one sensor no matter where object facing direction, full-view coverage was proposed by Wang and Cao [1]. An object is called full-view coverage if its face image is always covered by at least one camera no matter which object facing direction and the cameras viewing direction is sufficiently close to the object facing direction. However, number of camera sensors required for achieving full-view coverage in a network could be large, which leads to expensive cost to maintain the network. Along with full-view coverage model, Tseng et al. [2] independently introduced k-Angle coverage model to maximize number of objects to be covered using the least number of sensors. k-Angle guarantees to cover an object from k different viewpoints satisfying some angle constraints. While this coverage model can take advantages of full-view, it only uses a fixed number of camera sensor when monitoring an object, and thus conquers the problem of large number sensors in full-view. This coverage model can be very potential when applying to barrier coverage in WCSN. In this article, we study a new barrier coverage problem under k-Angle coverage model, but this coverage model has been fine-tuned to adapt for our problem (called multiple-view coverage) due to the sensor barriers surveillance mobile objects. Hereinafter, we called this problem as multiple view barrier coverage (MVBC), detail about MVBC problem as followed: Given a WCSN including camera sensor nodes deployed randomly in the interest of region, we verify whether a barrier of the camera sensor nodes can be formed which guarantees both detecting and recognizing unauthorized intruder. Furthermore, we devise a metric to estimate coverage quality of the achieved camera sensor barrier.

Estimating the quality of coverage of wireless sensor network is a fundamental problem, so is barrier coverage [3, 4]. Currently, in the tradition model of attenuation, when evaluating the coverage quality of a sensor toward a certain point, the coverage value formula is defined to be affected only by the distance from a sensor to the considered point, which may lead to several exceptional inconsistency regarding evaluate the quality of coverage, especially the camera sensor barriers are achieved from MVBC problem. Furthermore, while approaching the problems of MVBC, researchers attempt to analyse the problem with different settings of parameters. However, without proper metrics, it is insufficient to actually assess the effectiveness of different models themselves. Therefore, to the best of our knowledge, we are the first devising a more preferable attenuated model assessing the coverage quality of the sensor network toward a point in the field of interest, called differential coverage model. The model can later be generalized to evaluate the coverage of the sensor network on a line or a closed region. To formulate the coverage value regarding information from every direction of the intruder, it is necessary to consider its shape. In this article, without loose generality, shape of object is assumed as a circle. This geometrical shape is sufficient since the coverage is considered in the 2-D plane only, and the circle itself has parts of the circumference followed every possible direction, which help illustrate the devised model more thoroughly and clearly.

In conclusion, this study investigates two major tasks: first, analyzing the probability of successfully forming barrier coverage in WCSN under multiple view coverage model; second, evaluating the average quality of obtained barrier coverage over sensing field based. This problem is vital for optimizing the efficiency of sensor placement in sensor networks when camera sensor nodes are randomly scattered for achieving barrier coverage under multiple view coverage model in a large scale. For instance, camera sensors have to be scattered by plane or artillery if the region of interest is hostile or inaccessible or deficiency on time, manpower or funds prevents careful arrangement of every single sensor.

The main contributions of this article are as follows.

- Formulate multiple view barrier coverage problem in WCSNs.
- Propose an efficient method to deterministically verify if a monitored field can be achieved a multiple view barrier coverage in WCSNs by any given set of camera sensors.
- Evaluate the obtained barriers with the devised metric.
- Conduce extensive experiments in various scenarios to examine the result and the computation time of the proposed algorithm.
- Analyze and estimate the parameters effect on performance of proposed model.

The rest of the paper is organized as follow. Related works are presented in section 2. Section 3 formulates the multiple view barrier problem. Section 4 introduces proposed algorithm. Section 5 gives our experiments along with computational and comparative results as well as conclusion in section 6.

2. Related works

The concept of barrier coverage [5], was first introduced specifically for intruder detection applications in WSNs where sensing regions of sensor nodes form one or multiple barriers so that any intruder penetrating the region of interest will be detected. Due to its superiorities for security applications, barrier coverage has received attentions in recent years. The barrier coverage problems in WSNs can be categorized into two subproblems: one is finding penetration paths. A penetration path is continuous curve with arbitrary shape, go though one side to the other side of a sensor field; other is building intrusion barrier for detecting intrusion of a mobile object when it traverse from on side to the other side of the sensing field. The first problems have thoroughly been delved into many researches such as [6, 7, 8, 9, 10], the second ones mostly focused on critical condition analysis (e.g., sensor node density) and barrier construction for stationary sensors with omni-directional sensing coverage models. [11, 12, 13, 14, 15]. Directional sensing coverage models then were widely used such as camera, radar etc., and taken into consideration in coverage problems as well as in barrier coverage problems. [16, 17, 18, 19, 20]. Barrier coverage problems in WCSNs are much more complexed and challenging compared to those in traditional scalar WSNs [21, 22, 23, 24], because of WCSNs having unique features.

The authors [21] proposed a collaborative technique for face analysis in WCSNs with a dual objective of detecting the camera view closest to a frontal view of the object, and assessing angles between the face directional and all the camera views based on additional fusion of local angle estimates. To gather more information of the stable object, especially face recognition, full-view coverage was introduced in [1] by Wang et al. An object is full-view covered if its face is always a camera to cover it no matter which face direction and the angle between the camera's viewing direction and the object's facing direction is less than a predefined parameter θ. The authors proposed a method for full-view coverage verification on a sensing field. After that, they derived an estimation of the sensor density needed for full-view coverage in a random deployment. Based on this work, Wang et al. further studied the problem of constructing a camera barrier in [24]. They proposed a method to select camera sensors from an arbitrary deployment to form a camera barrier and then presented a technique for reducing the number of cameras used since there might be redundant cameras (cameras that can be turned off without breaking the barrier) after barrier is formed. Besides, Ma et al. [22], proposed a better method for constructing camera barrier. With aiming at minimum the number of camera sensors in full-view barrier coverage, the problem is transformed into the shortest path problem from the source to the destination node on graph. **However, the algorithm using in this work cannot be used to solve the problem**. This algorithm can be considered as a variant of Dijkstra algorithm. The only two differences are: (1) the edge has no weight but each vertex v has a "weight" denoted by $I(v)$ and (2) the operator used for updating label of a vertex is union instead of addition. The second difference is the reason why the algorithm has problem. According to the algorithm, when a vertex v is labeled, every previous vertex in the path from source vertex s to v is also labeled. Supposed u is such a vertex between s and v, the shortest path from s to u must be contained in the path from s to v. But this is not true when the operator union is used, Figure 1 illustrates this.

To monitor the object from multiple perspectives, Tseng et al. [2] introduced the notion of k-angle coverage. To avoid duplicating information from multiple sensors simultaneously monitoring an object, an angle constraint was added, which guaranteed any two sensors cannot appear in an angle range of ω around the object (Figure 2). It was pointed out that if an object is $(k-\omega)$ angle covered, there is no angle larger than $2\pi-(k-1)\omega$ of the object that is not covered by any sensor. This means that an object that is $(k-\omega)$ angle covered is also full-view covered with parameter $\theta=\frac{2\pi-(k-1)\omega}{2}$. Hence, $(k-\omega)$ angle coverage can be considered as a special case of full-view coverage with the number of camera sensors covering the object is fixed. Under this new coverage model, the paper focused on maximizing number of static objects that are covered using minimum number of sensors.

In [25], the authors studied the problem of constructing $(k-\omega)$-angle barrier using minimum number of sensors called MkABC. The paper presented MkABC problem in two sensor deployment schemes. Under deterministic deployment, a geometric method was proposed, which used the feature of regular polygon to construct a $(k-\frac{\pi}{k})$-angle barrier. When sensors are randomly deployed in the ROI, the MkABC becomes more difficult. In this scenario, the authors proposed a grid-based method, where each grid is judged to be $(k-\omega)$-angle covered or not. MkABC problem is then transformed into the shortest path problem on graph. The algorithm used is the same as one used in [22], which has some problems as aforementioned. Besides, The grid-based method has a trade-off between solution accuracy and the computational cost, which all depend on grid size chosen. This is a big disadvantage of this approach in large-scale WSN.

Chen et al. [26] first mentioned the problem of measuring the quality of barrier coverage in ODSNs. The authors introduced the notion of L-local k-barrier coverage to measure the quality of k-barrier coverage for a belt region as the maximum value of L that the belt is L-local k-barrier covered. A belt region is said to be L-local k-barrier covered if every zone of length L in the region is k-barrier covered. The measure always provides the same result when sensor network has already achieved k-barrier coverage, i.e, the probability of detecting the intruder by k sensors is always 100% which is equivalent to measuring its quality as 1 else zero, is not enough since there might be many different levels of quality coverage of the sensor barrier. In addition, the considered k-barrier is just combination of consecutive sensing range. Actually, these metrics reflect constructing level of k-barrier, i.e. the closer distance L and the length of the strip region is, the more ROI achieves k-barrier. In contrast, our purpose evaluates quality of collected information in camera sensor barriers. This prompts us to devise a novel metrics called Differential coverage.

After considering many related works, we see that previous researches about barrier coverage problems in WCSNs are not yet efficient and there are rooms for improvement. Basing on $(k-\omega)$-angle coverage model [25] with some fine-tuned for adapting to monitor mobile objects in barrier coverage in WCSNs, which refers to multiple view coverage model. Therefore, we produce the multiple view barrier coverage problem in WCSNs, then propose method as **Dynamic Partition** to solve this problem. Furthermore, we desire to measure the quality of object's information recorded by sensors network when it crosses the barrier. Since the metrics proposed in [26] is for k-barrier coverage model in ODSNs, it cannot apply to our problem. Moreover, this metrics only works when sensors network has not provided k-barrier coverage yet. In contrast, we need a metrics for measuring quality coverage of the barrier, which means the barrier must have been already constructed. These have fostered us to devise a new metrics called **Differentiation Exposure**.

3. Preliminaries and problem formulation

3.1. Preliminaries

Definition 1. *multiple-view coverage*

- A point P is multiple-view covered with k sensors and angle constraint ω if there exists a list of k sensors $L = \{S_1, S_2, ..., S_k\}$ ordered in counter-clockwise order around P, such that $\omega < (\overrightarrow{PS_i}, \overrightarrow{PS_{i+1}}) < \pi, \forall i \in \{1, 2, ..., k\}$ (consider $k+1 \equiv 1$)
- A region R is said to be multiple-view covered if every point in R is multiple-view covered.

Hereinafter, we use two concepts *multiple-view* coverage and (k, ω) coverage equivalently. With a specific value of k and ω, we always use (k, ω) instead of *multiple-view* (see figure 2)

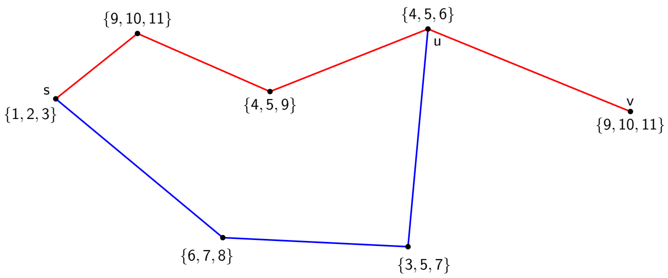


Figure 1: Red line is shortest path from s to v. Blue line is shortest path from s to u

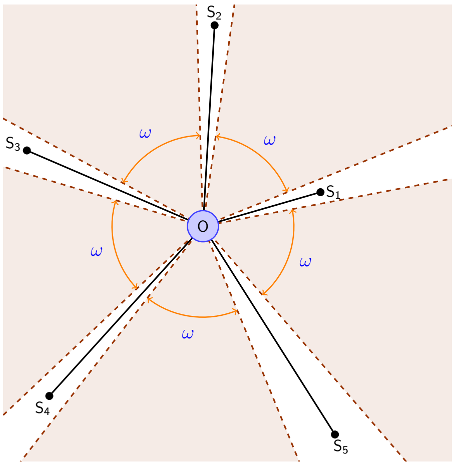


Figure 2: O is 5-ω-angle covered by $\{S_1, S_2, S_3, S_4, S_5\}$

Definition 2. *Safe region*

- Safe region was defined in [2]. Figure 3 illustrates the safe region of a line segment S_1S_2

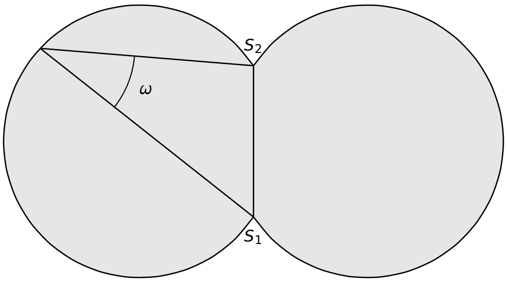


Figure 3: Safe region of line segment S_1S_2

Definition 3. *Inner safe region*

Inner safe region is attached to a side of a polygon, not an arbitrary line segment. Given a polygon POL. Let AB is a side of POL. The way to determine the inner safe region of AB is as follows:

- Choose the midpoint M of AB.
- Draw an arrow from M toward the inner region of POL.
- The safe region of AB consists 2 symmetrical parts splitted by line AB. The part that the arrow points to is the inner safe region of AB.

Figure 4 is an illustration of inner safe region of S_1S_2.

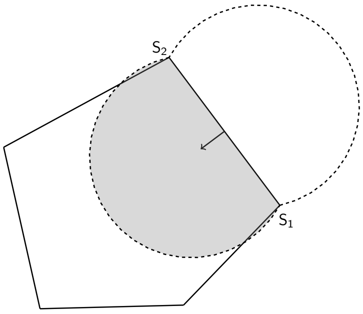


Figure 4: inner safe region of S_1S_2

Definition 4. (k, ω) *region of an* k*-sensor list*
Given a k-sensor list $L = \{S_1, S_2, ..., S_k\}$. (k, ω) region of L is the locus of points that are (k, ω) covered by L.

Theorem 1. (k, ω) *region of an* k*-sensor list* $L = \{S_1, S_2, ..., S_k\}$ *is the intersection of inner safe region of every line segment* S_iS_{i+1} *and the sensing range of all sensors in* L*.*

Proof. Let Σ denote the (k, ω) region of L. We only consider the case that Σ is not empty. Suppose that P is a point inside Σ, then P is (k, ω) covered by L. From definition 1, we have $(\overrightarrow{PS_i}, \overrightarrow{PS_{i+1}}) > \omega$ (1) and $(\overrightarrow{PS_i}, \overrightarrow{PS_{i+1}}) < \pi$ (2), $\forall i \in \{1, 2, ..., k\}$. Condition (1) means that P is inside the safe region of S_iS_{i+1}. This safe region has 2 symmetrical parts splitted by line S_iS_{i+1}. Condition (2) forces P to be located at only one special part which satisfies $(\overrightarrow{PS_i}, \overrightarrow{PS_{i+1}}) < \pi$. And this part is always the inner safe region of S_iS_{i+1}. Hence, P is inside the intersection of inner safe region of every line segment $S_iS_{i+1},\ i = \overline{1, k}$ and the sensing region of all sensors in L (call this intersection $\mho$) (*).

On the other hand, if P is inside $\mho$, P satisfies the condition (1) and (2). Thus, P is $(k - \omega)$ covered by L, which deduce to P is inside Σ (**).

From (*) and (**), we have $\Sigma \equiv \mho$ and theorem 1 is proved.

Theorem 2. *(k, ω) region of an k-sensor list $L = \{S_1, S_2, ..., S_k\}$ is a convex region.*

Proof. The intersection of two convex regions is a convex region. Hence, the intersection of any limited sets of convex region is a convex region. From theorem 1, it is obviously that (k, ω) region of a list L is intersections of convex regions and thus, we have theorem 2 proved.

Figure 5 is an illustration of this theorem. By that, the shadow area is the $(5, 60^o)$ region of $L = \{S_1, S_2, S_3, S_4, S_5\}$.

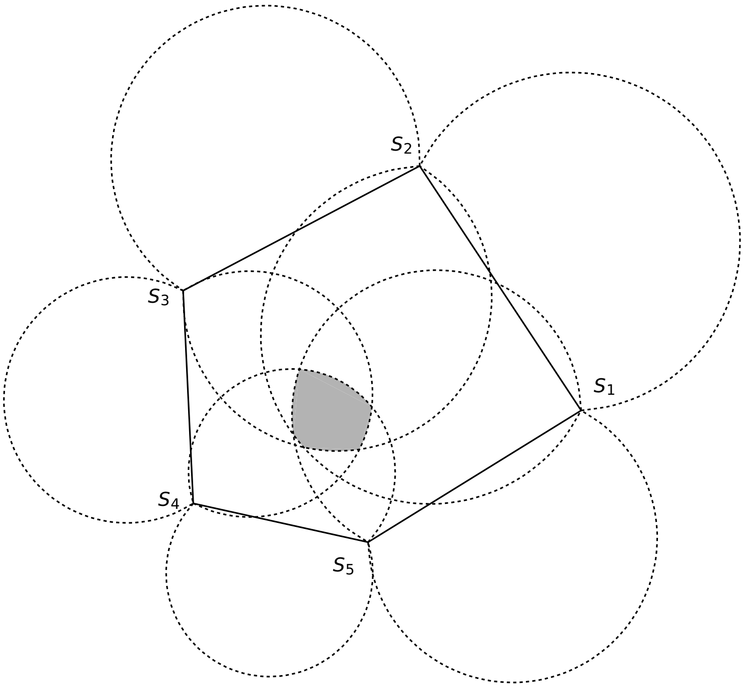


Figure 5: $(5, 60^o)$ region of $L = \{S_1, S_2, S_3, S_4, S_5\}$

Definition 5. *Differentiation Field Intensity*

In the tradition models to evaluate the coverage from the sensing field $S = S_1, S_2, ..., S_n$ toward a point P may lead to several exceptional inconsistency due of these model just depend on distance between sensors in sensing field and the considering point [6] as follows:

All sensing field intensity function

$$I_a(P) = \sum_{i=1}^{N} f(S_i, P) \tag{1}$$

Closet sensing field intensity

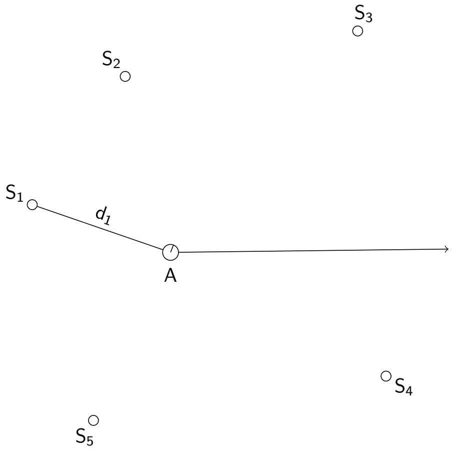
$$S_{\min}(P) = \underset{s \in S}{\text{argmax}} f(s, P) \tag{2}$$

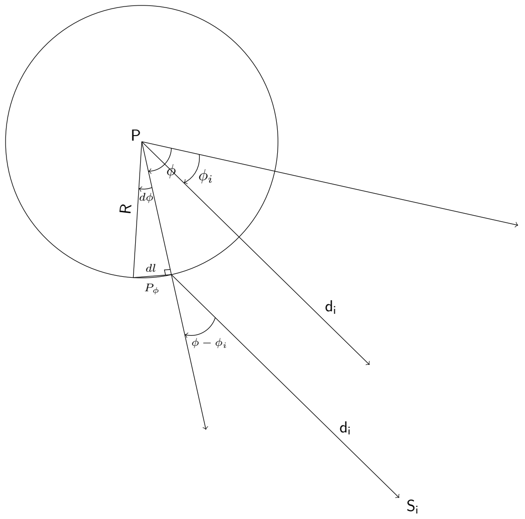
$$I_c(P) = f(S_{\min}(P), P) \tag{3}$$

where f is sensing intensity function that takes two arguments: the first is a sensor and the second is a point.

As a result, we devise a more preferable attenuated model assessing the coverage quality of the sensor network toward a point in the region of interest, the model can later be generalized to evaluate the coverage of the sensor network on a line or a closed region.

Considering a point P lie in the sensing range of a certain omnidirectional or directional sensor. The sensor S_i, the penetration object P with radius R and the considered part P_ϕ being positioned at ϕ and has length dl. Call the distance from S_i to P_ϕ as d_i, the direction of the sensor compared to the pivot direction as ϕ_i. Because R is usually inconsiderable compared to d_i, we can approximately use ϕ_i and d_i as constants with variable ϕ, the model will be now illustrated as followed.





Firstly, the coverage value of a sensor to a part is directly affected by the distance between the sensor and the part, and the angle at which the part is viewed by the sensor, this results in the formula:

$$\text{max}(\frac{A\cos(\phi - \phi_i)}{R d_i^\lambda}, 0) dl \tag{4}$$

where $\frac{A}{R}$ is a constant coefficient, note that the coverage would fall below 0 if the angle between the direction of the part and the direction of the sensor is larger than $\frac{\pi}{2}$, so we need to set it to 0 in that case. Rewrite $dl = R d\phi$, we have:

$$\text{max}\left\{\frac{A\cos(\phi - \phi_i)}{d_i^\lambda},\ 0\right\} d\phi \tag{5}$$

However, it is obviously unnecessary to obtain too much detailed information from the object in the sensing field. This leads to the existence of a constant E_{max} which corresponds to the maximum necessary coverage on a part with unit length of the circle. To isolate the value from the relative constant A, we rewrite it to the Minimum sensing radius $E_{\mathsf{max}} = \frac{A}{d^{\lambda}_{\mathsf{min}}}$. As a result, our coverage formula could be rewritten as:

$$\mathsf{max}\left\{0,\ \mathsf{min}\Big\{\frac{A}{d^{\lambda}_{\mathsf{min}}}, \frac{Acos(\phi - \phi_1)}{d_i^{\lambda}}\Big\}\right\} d\phi \tag{6}$$

As a result, the coverage on a part P_ϕ of several sensors S_i, as illustrated above, is the maximum of the coverage on that part of every covered sensor:

$$E_\phi(P) = \mathsf{max}\left\{0,\ \mathsf{min}\Big\{\frac{A}{d^{\lambda}_{\mathsf{min}}},\ \mathsf{max}_{S_i}\Big\{\frac{A\cos(\phi - \phi_i)}{d_i^{\lambda}}\Big\}\Big\}\right\} d\phi \tag{7}$$

In short, the coverage on a part of a certain set of sensors is calculated from the largest value of $\frac{Acos(\phi - \phi_1)}{d_i^{\lambda}} d\phi$ across all sensors, the result then will be a value in the close interval $[0, E_{\mathsf{max}} d\phi]$ that closest to the above computed value. The total coverage on the object is the sum of the coverage on its small parts. Combined with the differentiated form of the formula above, the total coverage would be the integral on all of its parts. As a result, we receive the formula for the total coverage on the object at a certain point in the sensing field:

$$I(P) = \int_0^{2\pi} \mathsf{max}\left\{0,\ \mathsf{min}\Big\{\frac{A}{d^{\lambda}_{\mathsf{min}}},\ \mathsf{max}_{S_i}\Big(\frac{A\cos(\phi - \phi_i)}{d_i^{\lambda}}\Big)\Big\}\right\} d\phi \tag{8}$$

In conclusion, a new model of coverage is devised which may prove to be exceptionally effective in measuring the coverage efficiency of sensor networks in not only the tradition coverage problem but also in more complex ones such as the problem of $full\ view$ or multiple view barrier coverage. The new model is proposed with detailed and precise logical progress, successfully adapts the strong points of both the All-Sensor Field Intensity and the Closest-Sensor Field Intensity model [6], handling preferably the cooperation of multiple sensors in the network without overrating the repetition of captured information.

Definition 6. *multiple-view barrier*
A multiple-view barrier B is a connected region from the left side to the right side of the monitoring region and satisfies that B is multiple-view covered.

Definition 7. *multiple-view barrier coverage*
A region achieves multiple-view barrier coverage if there exists a multiple-view barrier in that region.

A multiple-view barrier is a region connecting the left and right side of the sensing field in which all the points are multiple-view covered. Typically, a multiple-view barrier is fairly narrow, and penetration objects usually intersect the barrier only at a small part on their paths. As a result, a proper metric to assess the efficiency of the multiple-view barrier would be the coverage density of it.

With the same set of sensors considered, in the range of coverage of all elements of that set, the coverage function is always continuous. Since a barrier is consisted of several separate parts each of which is multiple-view covered by a common set of sensors, the coverage density of the barrier can be defined as the quotient of the total coverage in the barrier and the area of that area, with the total coverage being formulated as the integral of the sensing intensity function over the barrier region. Call the barrier region B with area S_B, the coverage density over B, which is D_B can be formulated as

$$I(B) = \iint_B E(x,y)dxdy.\frac{1}{S_B} \tag{9}$$

Table 1 summarizes all commonly used notations in this paper. Each notation has unique meaning throughout the whole paper.

Table 1: Commonly used notations

Notations	Description
Ω	The region of interest
L, W	Length and width of the region of interest respectively
n	Number of deployed camera sensors
S	Set of deployed camera sensors
S_i	i-th camera sensor in set S, also the location of S_i
s	An arbitrary sensor
R	Radius of every camera sensor
α	Half of the sensing range of every camera sensors
φ_i	Orientation angle of camera sensor S_i
k, ω	The conditional parameters of the problem
P	An arbitrary point in Ω, denotes an object
$f(s, P)$	Sensing intensity of sensor s towards P
$I(P)$	Sensing intensity of the whole set S towards P
Π	Set of multiple-view covered rectangles
B	A multiple-view barrier, actually a region consisting of connected rectangles
S_B	Area of the barrier B
$I(B)$	Coverage value of barrier B

3.2. *Problem formulation*

3.2.1. *Verify the $(k - \omega)$ barrier coverage*

The problem is formulated as follows. Given a set of n sensors $S = \{S_1, S_2, ..., S_n\}$ and a rectangular region Ω with the length of L and the width of W. Ω is called the monitoring region and camera sensors in S are deployed according to uniform deployment scheme in Ω to serve the purpose of observation. The uniform deployment scheme means that total n sensors are deployed randomly, uniformly and independently.

The objective of the problem is to verify if Ω achieves multiple-view barrier coverage. In other words, we need to determine if there exists a multiple-view barrier B in Ω. If there is none, Ω will not guarantee security requirements and the sensors need to be re-deployed.

Unlike omni-directional sensor, which only provides information about detection of the object, camera sensor is typically directional sensor and can be used to obtain multimedia information of the object. Each camera sensor can be denoted by a 4-tuple $\{S_i, R, \alpha, \varphi_i\}$, where S_i is the location of sensor i, R is the sensing radius and α is half of the sensing angle. We assume that all sensors have the same sensing radius and sensing angle. In reality, sensing range of camera sensor is usually less than π, so we also have an assumption that $\alpha < \frac{\pi}{2}$. The last parameter of a camera sensor, φ_i, is the facing direction of sensor i, which is uniformly distributed in $[0, 2\pi]$

Figure 6 shows information of sensor s.

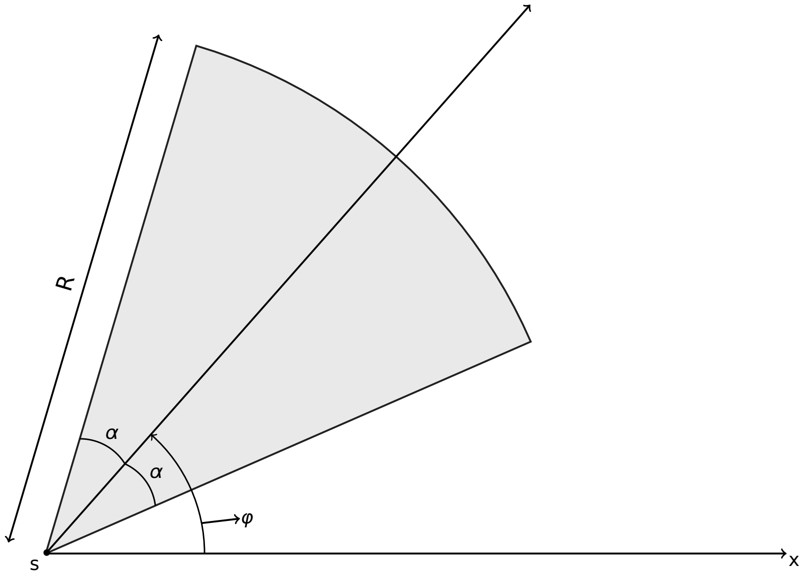


Figure 6: Illustration of a camera sensor s. The shadow area is the sensing region of s

The input and output of the problem are followings:

Input

- L, W: Length and width of the monitoring region Ω respectively.
- n: Number of camera sensors.
- $S = \{S_1, S_2, ..., S_n\}$: Set of camera sensors. S_i denotes the i-th camera sensor and also denotes the location of that sensor.
- R: Radius of camera sensors.
- α: Half of the sensing angle of camera sensors.
- φ_i: Orientation angle view of S_i where $i = \overline{1, n}$.
- k, ω : The conditional parameter of the problem.

Output

- The yes/no answer that the monitoring region achieves multiple-view barrier coverage.

3.2.2. Evaluate the quality of a multiple-view barrier

The problem is formulated as follows. Given a barrier B in a sensing field containing several connected regions B_i from the left to the right boundary of the field. Each B_i is a closing field that is $(k - \omega)$ covered by a k-list of sensors P_i.

The objective of the problem is to find the coverage of the barrier regarding our devised metric. The process is to assess the quality of the found barrier and compare the result with other settings of parameters to analyze the effect of each parameter to the quality of the sensing field and find the best combination of settings to achieve our desire.

The input and output of the problem are followings:

Input

- $\{B_i\}$: The set of closing region connect the left and the right edge of the sensing field.
- $\{P_i\}$: The set of k-list of sensors, the P_i is known to (k, ω) cover the region B_i.

Output

- The coverage value of the (k, ω) barrier.

4. Proposed algorithm

4.1. Verify the multiple-view barrier cover

To solve this problem, the monitoring region is partitioned into several small rectangles using the proposed Dynamic Partition method. After that, we try to figure out if there is a continuous barrier from the left side to the right side of the region consisting of rectangles which are multiple-view covered. The details are shown in the subsequent sections.

4.1.1. multiple-view verification on a rectangle

Based on theorem 2, we can conclude that in order to verify the multiple-view coverage on a rectangle, it is sufficient to check whether all four vertices of that rectangle is multiple-view covered by an common list of sensors. Using this conclusion, we propose an algorithm to verify multiple-view coverage on a rectangle, and for the optimization of the second problem, we try to find the list that multiple-view cover the rectangle with the largest coverage value. The following steps describes the idea to verify if rectangle $ABCD$ is (k, ω) covered:

Step 1: Find a set of sensors G that cover four vertices of the rectangle $ABCD$.

Step 2: Find all lists of k sensors from G satisfies that the point A is (k, ω) covered by these k sensors.

Step 3: Among found lists, filter out those which do not simultaneously (k, ω) cover three points B, C, D. This step will offer all lists of sensors that (k, ω) cover the rectangle $ABCD$. If there does not exist such a list, $ABCD$ is not (k, ω) covered. Otherwise, go to **Step 4**.

Step 4: From lists found in **Step 3**, select one to later perform coverage quality evaluation on. There are 2 ways to choose the appropriate list among the satisfying lists found, which will later be called node handling method:

- Max method: The chosen list would be the one which offers the greatest coverage value toward the considered node.
- Random method: A random list is chosen, this list is expected to reveal the expectation of coverage value of all the sensor lists that $(k - \omega)$ cover the considered node.

Note that to verify (k, ω) coverage on $ABCD$, it is sufficient to stop at **Step 3**. **Step 4** is necessary to perform evaluation of quality coverage on $ABCD$. Criteria for selection in **Step 4** is called *node handling method*. In this paper, we consider two node handling methods, which are the max and the random ones. The max method chooses the list which offers the highest coverage value. The random one picks a list randomly, which is expected to give the expectation of coverage value of all lists that (k, ω) covered the rectangle. Different node handling methods are considered to guarantee that our algorithm is convergent and stable in different scenarios (detailed results can be found later in section 5)

The key to implement this idea is at **Step 2**. Our approach to this problem is very natural. First, sort G in counter-clockwise order around A. Then, we consider each sensor in G sequentially. If the sensor being considered satisfies some conditions, we put it into a list (call this list L). We do that until size of L is equal to k. Then, L is called a valid list. Figure 7 illustrates how to choose a valid list. In figure 7, black vector denotes the sensor that is chosen to put into the list, while red vector denotes candidates to be chosen.

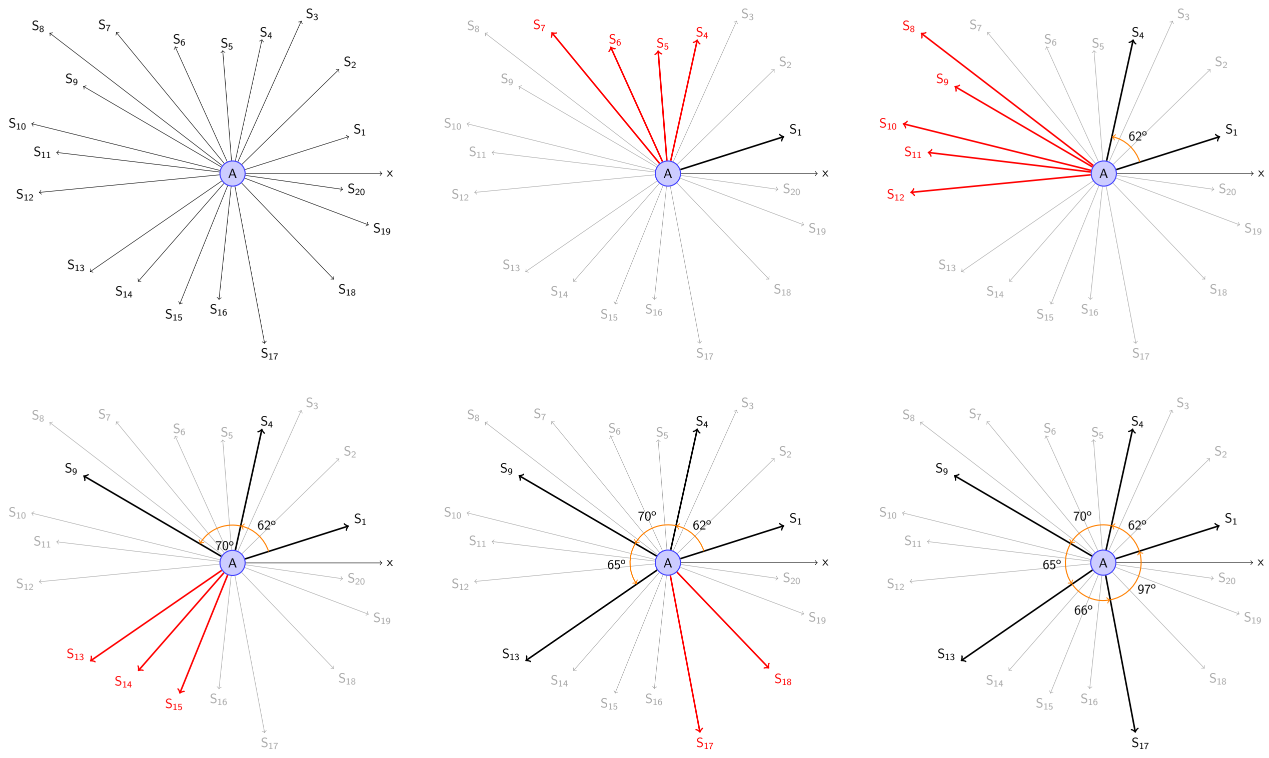


Figure 7: $L = \{S_1, S_4, S_9, S_{13}, S_{17}\}$ is a valid list with $k = 5, \omega = 60^o$

As aforementioned, when considering a sensor in G, it must satisfies some conditions to become candidate to be put into the list L. Suppose that at some point of the finding process, the list has *index* elements and *L[index]* $=$ *G[current]*, *1* $\leq$ *current* $\leq n$, n is size of G. If *G[next]* is chosen to be the next element in L, it must satisfy two conditions:

- $(\overrightarrow{PG[\text{cur}]}, \overrightarrow{PG[\text{next}]}) > \omega \qquad (i)$
- $(\overrightarrow{PG[\text{next}]}, \overrightarrow{PL[1]}) > (k - \text{index})\omega \qquad (ii)$

From definition of (k, ω) coverage, condition (i) is clearly necessary. However, it's not sufficient for *G[next]* to become a candidate for the next position in L. If L is a valid list, we have $(\overrightarrow{PL[i]}, \overrightarrow{PL[i+1]}) > \omega, i = \overline{1, k}$

(consider $k+1=1$). Hence, $(\overrightarrow{PL[\text{index}+1]}, \overrightarrow{PL[1]}) = \sum_{i=\text{index}+1}^{k} (\overrightarrow{PL[i]}, \overrightarrow{PL[i+1]}) > (k-\text{index})\omega$. Since we are choosing candidate for (index + 1)-th element in L, G[next] corresponds to L[index + 1]. Thus, (ii) is also a necessary condition.

Algorithm 1 and 2 show the details of our method. Algorithm 2 is a support function for Algorithm 1. `RecurFinding(cur)` is a recursive function that finds candidate for (index + 1)-th position in L knowing that there is a set cur containing the chosen sensors, in which the last sensor has index last. It considers elements in G sequentially from (last + 1)-th element and checks if these elements satisfy condition (i) and (ii). The return value of `RecurFinding(cur)` is a set containing all the lists of sensors that $(k-\omega)$ cover the point P which takes the current sublist $(\{L[1], L[1], ..., L[\text{index}]\})$ as its first index elements exists. This return value is used to support the recursion process of the algorithm.

```
Algorithm 1: Find all lists of $k$ sensors that $(k-\omega)$ covers point $P$

  Input:  A point P and a set G consisting of n
          sensors that cover P.
  Output: k sensors that $(k-\omega)$ covers P.
          There is possibility that no output is
          found.

Let L store the output
Sort G in counter-clockwise order around P
$L \leftarrow \emptyset$
for $i = 1$ to $n$ do
    $temp \leftarrow$ RecurFinding(Gi)
    $L \leftarrow L \cup temp$
end for
```

```
Algorithm 2: Find index + 1 element in L

  Input:  A list of sensors that is currently
          chosen, contains index sensors.
  Output: All lists of k sensors that $(k-\omega)$
          cover the point P beginning with the
          input list.

RecurFinding(cur)
    $L \leftarrow \emptyset$
    index $\leftarrow$ cardinality of cur
    if $m == k$ then
        return {cur}
    end if
    $last \leftarrow$ index of last element in cur
    $first \leftarrow$ index of first element in cur
    for $i = last + 1$ to n do
        if $(\overrightarrow{PG[last]}, \overrightarrow{PG[i]}) > \omega$ &&
        $(\overrightarrow{PG[i]}, \overrightarrow{PG[first]}) > (k - \text{index} + 1)\omega$
        then
            $temp \leftarrow$ RecurFinding(cur +
              (G[i]))
            $L \leftarrow L \cup temp$
        end if
    end for
    return L
end
```

The first element of L can be any sensor in G since it doesn't require any condition. For convenient, we choose $L[1] = G[1]$. Thus, `RecurFinding({G[1]})` is called to start the finding process. After function call `RecurFinding({G[1]})`, the function will return all the satisfied lists containing G[1]. The finding process stops when we have called `RecurFinding({G[i]})` with every i from 1 to n. And the algorithm will output a set containing all the lists of sensors that $(k-\omega)$ cover the considered point P.

4.1.2. Finding a barrier in a monitoring region

a, Partitioning the monitoring region by Dynamic Partition method

To find a barrier in the monitoring region, we first determine the areas that are multiple-view covered inside the region of interest Ω. To solve this problem, we partition Ω into multiple small rectangles and check whether these rectangles are multiple-view covered or not. However, uniform partitioning often requires a high computation time especially when the Ω is large. To overcome this challenge, we propose a new partition method called Dynamic Partition. The idea of the Dynamic Partition method is as follows: only the rectangles which are not multiple-view covered will be partitioned into smaller rectangles, otherwise, they are kept untouched.

The first rectangle to be checked is the monitoring region. Using the algorithm in 4.1.1, if a rectangle is multiple-view covered, mark it as true, otherwise, split it into four equal sub-rectangles. After a rectangle is split, smaller rectangles are generated and the process of checking and splitting is applied to these new rectangles. A rectangle will not be split if it is multiple-view covered or its size reaches a predefined limited value. The smaller the limited size is, the more precise the result of our algorithm can get. This condition guarantees our algorithm not to go into an infinite loop. The process is illustrated in Figure 8.

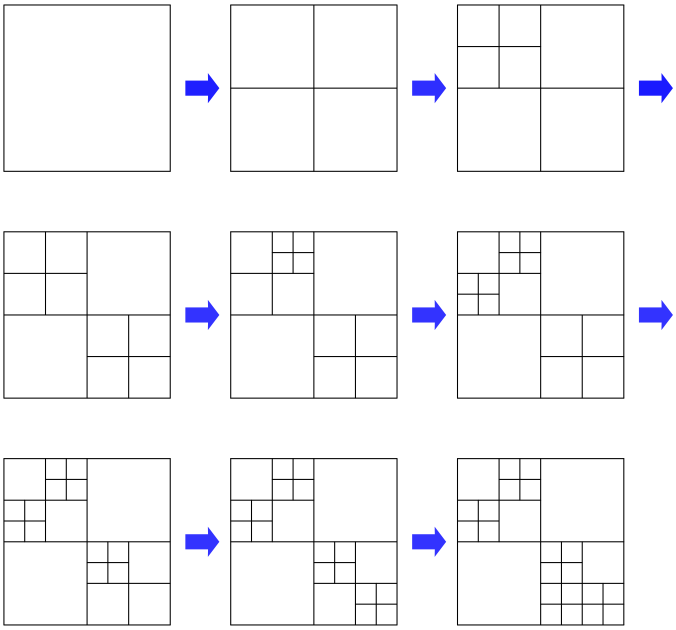


Figure 8: Illustration of Dynamic Partitioning

The pseudo code of the method is described in Algorithm 3

Algorithm 3: Dynamic Partition

```
Input:
  • L, W: Length and width of the monitoring region $\Omega$ respectively
  • A set of deployed sensors $S$
  • Minimum size of a grid
Output: A set of multiple-view covered rectangles $\Pi$

Let rootRec denote the monitoring region
Q $\leftarrow \varnothing$
enqueue(rootRec, Q)
while Q $\neq \varnothing$ do
    tempRec $\leftarrow$ dequeue(Q)
    if tempRec is multiple-view covered then
        add tempRec to $\Pi$
    else if tempRec has not reached the minimum size then
        split tempRec into 4 sub-rectangles
        add 4 sub-rectangles of tempRec to Q
    end if
end while
```

b, Finding a multiple-view coverage barrier

After procedure in 4.1.1, we now have a set Π of rectangles that are multiple-view covered. To find a multiple-view covered barrier, we need to find a continuous area formed from rectangles in Π that connects the left side to the right side of Ω. The method is to transform the rectangles set into a graph. Each vertex in the graph corresponds to a rectangle in Π. Two vertices are considered adjacent if the corresponding rectangles share at least one point. Two virtual vertices are added to the graph, source vertex `source` and sink vertex `sink`. All vertices corresponding to the rectangles lying on the left side of Ω are adjacent to `source` and all vertices corresponding to the rectangles lying on the right side of Ω are adjacent to `sink`. After constructing the graph, we use Breath First Search algorithm to find a path from `source` to `sink`. If a path is found, we conclude that there exists a multiple-view barrier in the monitoring region. Otherwise, the barrier does not exist.

4.2. Evaluate the quality of a multiple-view barrier

To assess quality of coverage of achieve sensor barrier, the coverage model implements the idea of Divide-And-Conquer. The object is differentiated into several small parts. Each of them is then evaluated separately, then added together to obtain the total coverage of the sensor network on the intruder at the considered position. Each small part of the circle, in turn, is calculated with each sensor that cover it, then the largest coverage value is taken. This operation will prevent the coverage overrated from several sensors having similar position toward the considered point, which in real life will serve no purpose of obtaining more information of the specific part on the intruder.

The algorithm takes the nodes forming a barrier and the k-list of sensors associating with each node as the input and compute the coverage on the input barrier.

The coverage of the barrier is calculated as the average of every node which forms that barrier with the weight assigned as the area of each node. With B_i as the nodes forming the barrier B, we have

$$E(B) = \iint_B E(P)dxdy.\frac{1}{S_B}$$

$$= (\sum_i (\iint_{B_i} E(P)dxdy)).\frac{1}{S_B}$$

$$= (\sum_i E(B_i).S_{B_i}).\frac{1}{S_B}$$

$$= (\sum_i E(B_i).S_{B_i}).\frac{1}{\sum_i S_{B_i}}$$

As a result, this calculation method is consistent with our definition of coverage on the barrier in the **??** section, hence may provide preferable assessment on each setting of parameters.

Since it is impossible and unnecessary to compute the exact coverage value of each note, it is sufficient to publish a method to estimate an approximation of the coverage on the considered node. Take into account the fact that in each node is $(k - \omega)$ covered by an unique k-list of sensors, hence the coverage value inside the node is a continuous function. As a result, we can create a dense grid in each node, and estimate the node coverage with the average of the vertices on the discrete grid. For convenience, the size of the grid is fixed to be the size of the node which will not be split further in 4.1.2.

5. Experimental results

5.1. Simulation method

This part will analyze the effect of several parameters on 3 aspects of the result, which is the probability of creating barrier, the average coverage of the multiple view sensor barrier and the overall computational time. The algorithm is performed on every instance and keep recording 3 data, the creation of barrier, the computational time and the exposure on a barrier if there is one. Then, the result are combined for all instances of the same parameter settings to achieve the probability of barrier creation, the average computational time and the average exposure value on the found barriers.

5.2. System setting and parameters setting

System settings

All the experiments are performed on a personal computer with core Intel Core i7-7700HQ, 8GB of DDR4 RAM running on Windows 10 Home, the programming language used to simulate the algorithm is Java 11.

Parameter settings

The sensing fields in all experiments are presented as rectangles with the size of 200m x 50m. Sensor nodes are deployed uniformly in a rectangle with each side extended compared to the sides of the sensing field a distance equal to the sensing radius of each sensor in order to guarantee the uniform distribution regarding sensing area inside the sensing field. Each set of parameters contains several independent random topologies to conduct the algorithm on and measure the target indexes. Furthermore, each instance of experiment is conducted with both node handling methods. Altogether there are 42000 experiments on 210 instances of parameters which were analyzed with our algorithm. The details are as follows:

Length	200
Width	50
Sensing Radius	30
Minimum sensing radius	5
Sensing angle	90
k	3, 4, 5, 6
The number of topologies for each instance	100
Node handling method	Max, Random

Table 2: General parameters

k	3	4	5	6
ω	90 - 115	55 - 80	40 - 60	35 - 50

Table 3: Problem parameters

5.3. Computation results

5.3.1. Effect of ω on algorithm performance

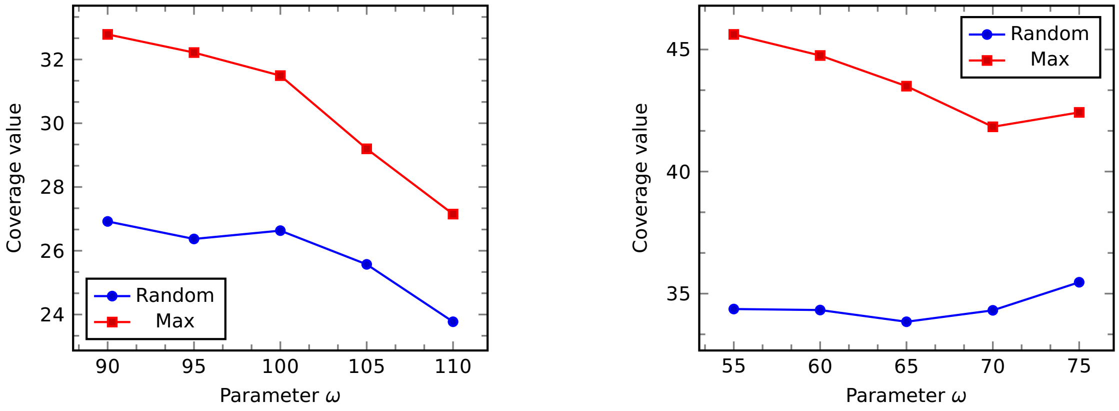
ω is an important parameter in the $k - \omega$ coverage model. As a result, this parameter has a considerable impact on the output of the algorithm. Because ω is a lower bound for the angle between two consecutive sensors in the perspective of the considered point, every sensor set that satisfies the condition with large ω would also successfully make a $k - \omega$ cover with lower ω. In short, a decrease in parameter ω may result in an expansion in the result space of the algorithm. This leads to two different consequences. On the one hand, there would be more sets of sensor $k - \omega$ cover a single node, which means that the coverage value of that node is likely to be lifted. However, on the other hand, the lower value of ω could reduce the average rank of the covered nodes, as the nodes are more easily covered, which leads to a lower coverage value, since the sets that cover the bigger node tend to position further than the sets covering the smaller ones.

As a consequence, firstly, with a lower value of ω, the algorithm would offer a greater chance of $(k - \omega)$ barrier existence. However, the probability of forming barriers can never exceed 100%, the curve that represents the barrier probability would approach 100% and does not rise higher with lower ω. Regarding coverage value, while the max method illustrate an downward trend, the random method tends to go up while the value of ω keep rising. This result is not applied to very large value of ω, where the low barrier probability leads to little number of examined sensing field, which results in a high variance and the obtained results would be less concrete. However, despite overall trends, the coverage value obtained with different ω is observed not to change significantly, and the difference can be accepted in real circumstances.

Finally, a lower value of ω results in a larger searching space, which leads to a drastic rise in computation time to conduct the max method, while the random method does not suffer from this property. This result may come from the fact that the max method require the evaluation of the coverage value of every $(k - \omega)$ sensor list toward each node, which seems to be an exception job.

5.3.2. Effect of sensor number on algorithm performance

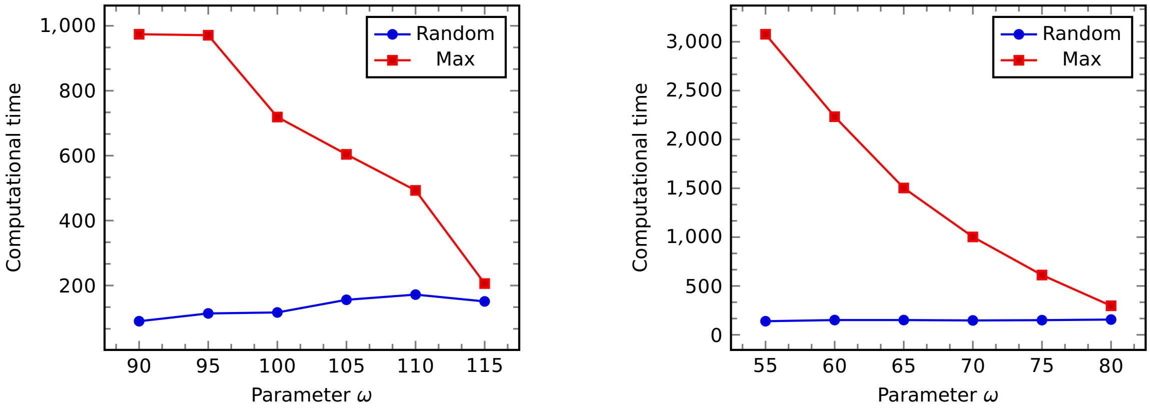
Similar to the effect of ω, a larger value of sensor number would lead to a larger searching space. However, in this occasion, the negative effect on coverage value seems to be more significant. As a result, the barrier coverage value tends to fall slowly as the sensor number rises for both node handling methods. Furthermore, the large number of sensors leads to a huge computational work, as the time required to run both methods rise linearly with the increase of the sensor number. This results in the computation time surge dramatically as the sensor number grows.



(a) k = 3

(b) k = 4

Figure 9: Effect of ω on the average coverage value of the found barrier with $k = 3$ and $k = 4$



(a) k = 3

(b) k = 4

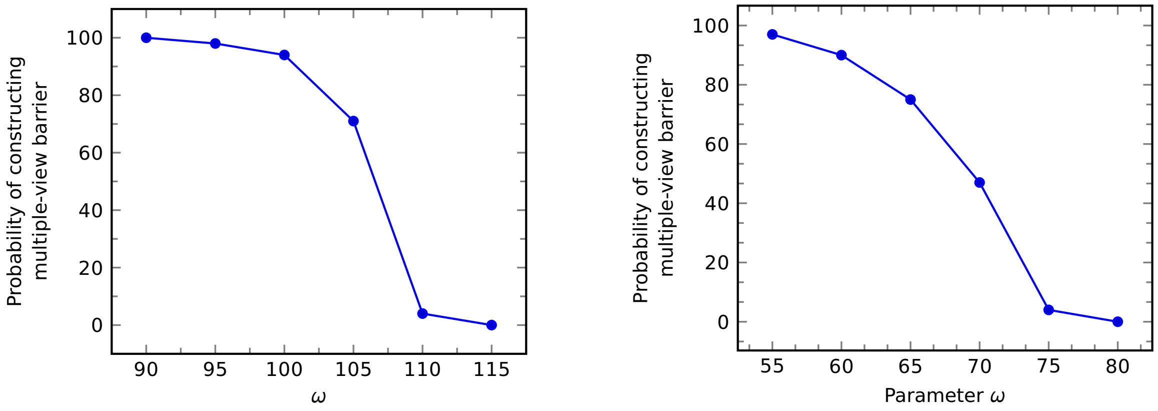
Figure 10: Effect of ω on the average computational time in ms of the algorithm with $k = 3$ and $k = 4$

5.3.3. *Effect of k on algorithm performance*

The parameter k is affected the achieved results the most regarding all 3 aspects. This is because the change in k would manipulate the problem entirely, an answer with a value of k would not be an answer with another value of k. As a consequence, the achieved results are drastically different among every value of k.

As mentioned in previous parts, generally, the coverage value of the barriers would not be much different from the others. As a result, we may reach a conclusion that for every value of k, it is possible to define a critical value of barrier coverage which denotes the largest achieved value of coverage for a certain value of k. And this critical coverage value could be use to compare the performance of the problem with different values of k.

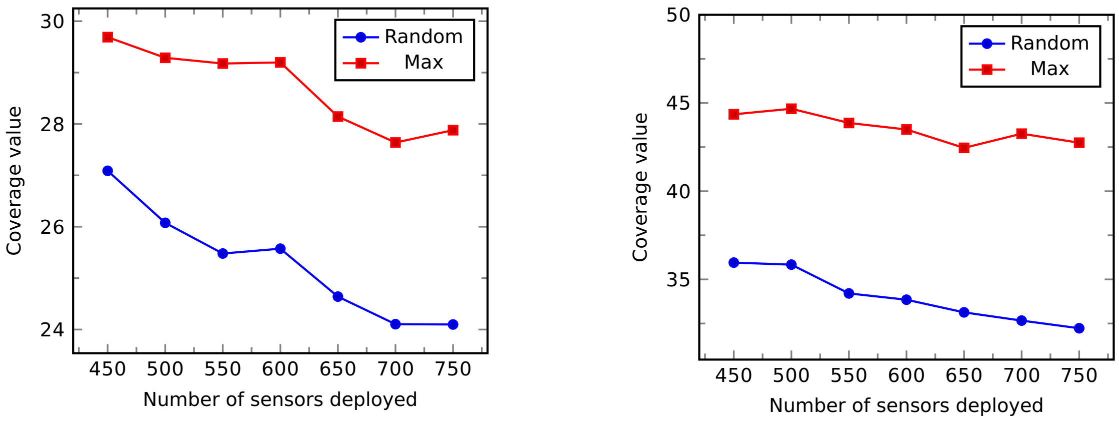
Regarding this metric, in general, as there are more sensors that cover a certain point, an increase in the value of k may lead to a larger critical coverage value. However, since the function of $cos(x)$ has a derivative getting lower as the value of x comes close to 0, and the effect of increasing k on decreasing the sight angle of sensors to the parts of the intruder $(\phi - \phi_i)$ may reduce with larger k. As a result, the critical coverage



(a) k = 3

(b) k = 4

Figure 11: Effect of ω on the probability of existence of multiple view barrier with $k = 3$ and $k = 4$



(a) k = 3, ω = 105

(b) k = 4, ω = 65

Figure 12: Effect of sensor number on the average coverage value of the found barrier with some typical values of k and ω

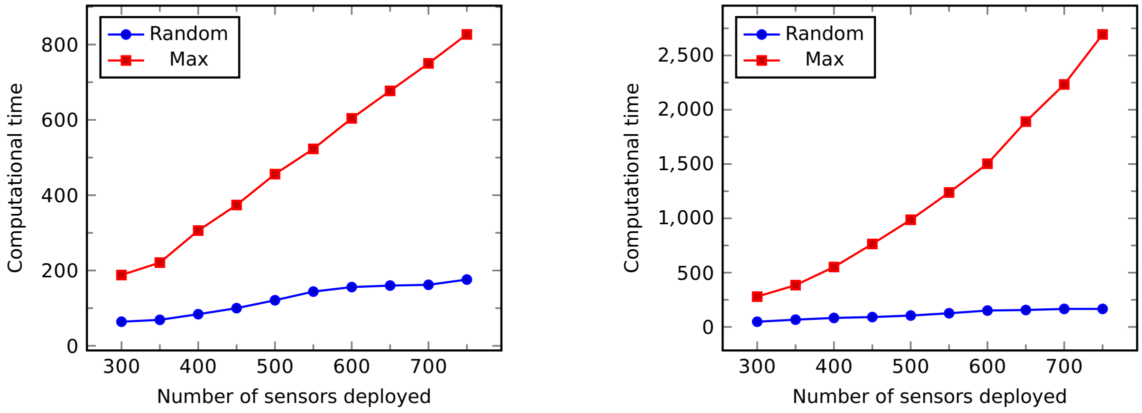
value would rise slower than the rise in the value of k.

Finally, the effect of k on computation time is hard to illustrate clearly, since each k would correspond to different values of ω, which makes it impossible to choose which settings of parameters should be compared. However, in general, a greater amount of k would lead to a significant greater computational time. This is because that the large value of k would leads to a larger nest in traversing for all the $k-\omega$ sets and larger loop when checking the coverage value of nodes, hence the computation time for finding all the sets that $k - \omega$ cover each node and determining the sets of sensor with largest coverage value is increased considerably.

5.3.4. Experimental conclusion

Regarding all above analysis, it is possible to achieve some valuable conclusion about the model of $(k-\omega)$.

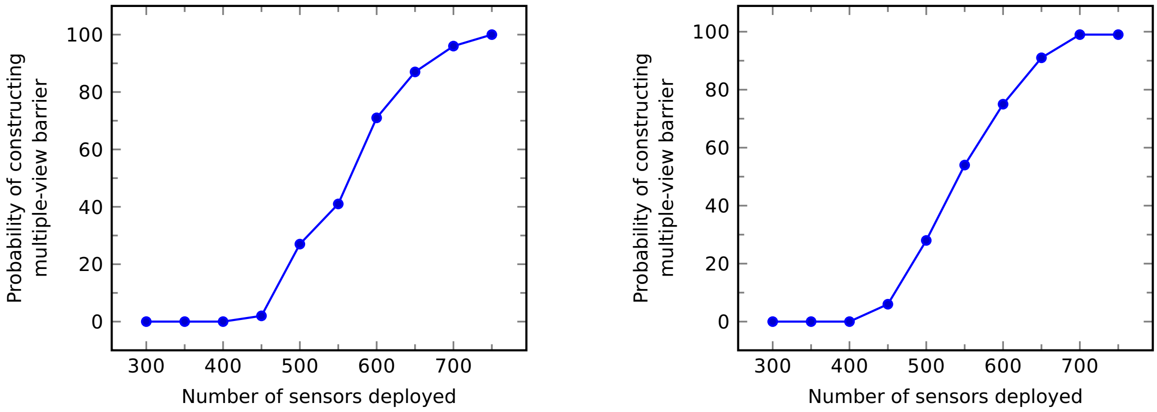
- Firstly, it is notable that the max method overcomes the random method regarding coverage value with a considerable gap between the method results. Despite the surge in computational time due



(a) k = 3, ω = 105

(b) k = 4, ω = 65

Figure 13: Effect of sensor number on the average computational time in ms of the algorithm with some typical values of k and ω



(a) k = 3, ω = 105

(b) k = 4, ω = 65

Figure 14: Effect of sensor number on the probability of existence of multiple view barrier with some typical values of k and ω

to expensive node handling, the significant offered coverage value prove that it is more suitable to choose the max node handling method.

- In terms of model parameters k, ω and sensor number, with the choose of max method, the model offer great result regarding coverage value, computational time and barrier probability with some appropriate settings of parameters.

6. Conclusion

This paper addressed the minimal exposure path problem for attenuated sensing model with all mobile sensors. We first have considered and formulated a model of the minimal exposure path problem in all mobile sensor networks; we have proposed the genetic algorithm to solve this issue. In addition, extensive experimental simulations were conducted to validate and to evaluate the proposed model as well as algorithm. The results showed that: the proposed algorithm could be effectively applied to both static and

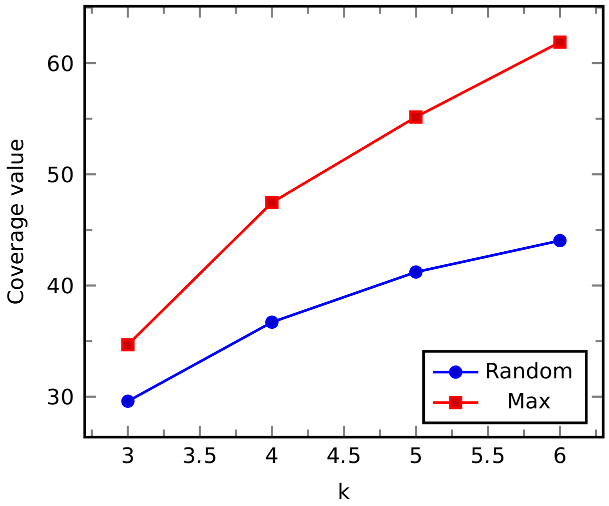


Figure 15: Effect of k on the critical coverage value of the $(k - \omega)$ barrier

mobile models of wireless sensor networks; the coverage of mobile sensor networks is almost better than the coverage of static sensor networks in case having the same number of sensors.

References

[1] Y. Wang, G. Cao., Achieving full-view coverage in camera sensor networks, ACM Transactions on Sensor Networks (ToSN) 10 (1) (2013) 3.

[2] Y. C. Tseng, P. Y. Chen, W. T. Chen., k-angle object coverage problem in a wireless sensor network, IEEE Sensors Journal 12 (12) (2012) 3408–3416.

[3] A. Sangwan, R. P. Singh, Survey on coverage problems in wireless sensor networks, Wireless Personal Communications 80 (4) (2015) 1475–1500.

[4] A. Ghosh, S. K. Das, Coverage and connectivity issues in wireless sensor networks: A survey, Pervasive and Mobile Computing 4 (3) (2008) 303–334.

[5] S.Kumar, T. H. Lai, A. Arora., Barrier coverage with wireless sensors, Proceedings of the 11th annual international conference on Mobile computing and networking (2005) 284–298.

[6] S. Megerian, F. Koushanfar, G. Qu, G. Veltri, M. Potkonjak., Exposure in wireless sensor networks: theory and practical solutionl, Wireless Networks 8 (5) (2002) 443–454.

[7] H. T. T. Binh, N. T. M. Binh, N. H. Hoang, P. A. Tu., Heuristic algorithm for finding maximal breach path in wireless sensor network with omnidirectional sensors, IEEE Region 10 Humanitarian Technology Conference (R10-HTC) (2016) 1–6.

[8] L. Liu, X. Zhang, H. Ma, Percolation theory-based exposure-path prevention for wireless sensor networks coverage in internet of things, IEEE Sensors Journal 13 (10) (2013) 3625–3636.

[9] N. T. M. Binh, C. M. Thang, N. D. Nghia, H. T. T. Binh., Genetic algorithm for solving minimal exposure path in mobile sensor networks, IEEE Symposium Series on Computational Intelligence (SSCI) (2017) 1–8.

[10] H. T. T. Binh, N. T. M. Binh, N. H. Ngoc, D. T. H.Ly, N. D. Nghia., Efficient approximation approaches to minimal exposure path problem in probabilistic coverage model for wireless sensor networks, Applied Soft Computing 76 (2019) 726–743.

[11] B. Liu, O. Dousse, J. Wang, W. Saipulla., Strong barrier coverage of wireless sensor networks, Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing (2008) 411–420.

[12] A. Saipulla, B. Liu, J. Wang., Barrier coverage with airdropped wireless sensors, MILCOM -IEEE Military Communications Conference (2008) 1–7.

[13] J. He, H. Shi., A distributed algorithm for finding maximum barrier coverage in wireless sensor networks, IEEE Global Telecommunications Conference GLOBECOM 2010 (2010) 1–5.

[14] P. Skraba, L. Guibas, Energy efficient intrusion detection in camera sensor networks, International Conference on Distributed Computing in Sensor Systems (2007) 309–323.

[15] J. Chen, J. Li, T. H. Lai., Energy-efficient intrusion detection with a barrier of probabilistic sensors: Global and local, IEEE Transactions on Wireless Communications 12 (9) (2013) 4742–4755.

[16] Coverage by directional sensors in randomly deployed wireless sensor networks, Journal of Combinatorial Optimization 11.

[17] I. Akyildiz, T. Melodia, K. Chowdhury., A survey on wireless multimedia sensor networks, Computer networks 51 (4) (2007) 921–960.

[18] M. Guvensan, A. G. Yavuz., On coverage issues in directional sensor networks: A survey, Ad Hoc Networks 9 (7) (2011) 1238–1255.

[19] H. Ma, Y. Liu., On coverage problems of directional sensor networks, International Conference on Mobile Ad-Hoc and Sensor Networks (2005) 721–731.

[20] S. Soro, W. Heinzelman., On the coverage problem in video-based wireless sensor networks, 2nd International Conference on Broadband Networks (2005) 932–939.

[21] C. C. Chang, H. Aghajan., Collaborative face orientation detection in wireless image sensor networks, Proceedings of ACM SenSys Workshop on Distributed Smart Cameras (2006).

[22] H. Ma, M. Yang, D. Li, Y. Hong, W. Chen., Minimum camera barrier coverage in wireless camera sensor networks, Proceedings IEEE INFOCOM (2012) 217–225.

[23] A. Makhoul, R. Saadi, C. Pham, Adaptive scheduling of wireless video sensor nodes for surveillance applications, Proceedings of the 4th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks (2009) 54–60.

[24] Y. Wang, G. Cao., Barrier coverage in camera sensor networks, Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing (2011) 12.

[25] B. Xui, Y. Zhu, D. Li, D. Kim, W. Wi., Minimum (k, ω)-angle barrier coverage in wireless camera sensor networks., IJSNet 21 (3) (2016) 179–188.

[26] A. Chen, T. H. Lai, D. Xuan., Measuring and guaranteeing quality of barrier-coverage in wireless sensor networks, Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing (2008) 421–430.