Intrusion Object Detection of a Randomized Scheduling Algorithm inWireless Sensor Network

Quangang Zhao
Dezhou University
Dezhou 253023, P.R.China
China
dzzhqg@gmail.com

ABSTRACT: *In wireless sensor networks, which can be configured to put some sensor nodes in sleep mode to save energy. This approach is a special case of a randomized scheduling algorithm, in which k subsets of sensors work alternatively. In this paper, we first study the performance of the randomized scheduling algorithm via both analysis and simulation in terms of intrusion coverage intensity, and detection probability, then, we also study the asymptotic coverage and other properties. Finally, we analyze a problem of maximizing network lifetime under Quality of Service constraints such as bounded detection delay, detection probability, and network coverage intensity. We prove that the optimal solution exists, and provide conditions of the existence of the optimal solutions.*

Keywords: Wireless Sensor Network, Quality of Service, Coverage, Optimization

Received: 7 October 2016, Revised 9 November 2016, Accepted 15 November 2016

1. Introduction

Wireless sensor networks (WSNs) have become an important technology, combining sensing technology, embedded computing, distributed information processing, and wireless communication technology [1] [2] [3]. So WSNs have broad applications [4] [5]. And sensor nodes are usually deployed in hostile environments. As a result, nodes and communication links are prone to failure. In the early centralized algorithms is used, but which is undesirable in sensor networks using resource-limited sensor nodes. In contrast, localized distributed algorithms are welcome to be used because they are simple and robust to network topology.

As for cooperative processing in a sensor network, the information of interest is aggregate statistics amid a group of sensor

nodes. So emerging models of sensor networks are more constrained than general models of distributed systems, For instance, the function $f(v) = \sum c_i f_i(v_i)$ is the sum aggregate of values $c_i f_i(v_i)$ which are pre-processed from v_i on all nodes.

Unlike previous work, in this paper, we present randomized algorithms (RA) for these problems and measure their complexity in the comparison model [6]. In that model, the algorithms use random bits to help determine which comparisons will be made, and RA is more efficient than gossip-based algorithms like Uniform Gossip. Because RA can works correctly and efficiently on all topologies, and which take advantage that can all nodes hear and receive a wireless transmission within the radio coverage.We suggest a modified broadcast based flooding to mitigate this pitfall and compare it with AR by simulations.

Because the research work shown that tree based algorithms face challenges in efficiently maintaining resilience to topology changes. Then, the authors of sensor tech report [1] build an optimal aggregation tree to efficiently computed the aggregates,which have addressed the importance and advantages of in-network aggregation. Now, we know that the minimum steiner tree problem problem is *NP - Hard*. Althougha a distributed heuristic tree approach could save the cost of coordination at the tree construction stage, the aggregation tree will need to be reconstructed whenever the topology changes, before aggregate computation can resume or re-start. On the other hand, distributed localized algorithms such as our proposed RA, Gossip algorithm of Boyd et al. Hence, distributed localized algorithms are more robust to frequent topology change in a wireless sensor network.

In contrast to conventional deterministic uncertain system models, the stochastic uncertain system models may be substantially richer, and converge with all nodes knowing the aggregate computation results. So, the computed results become robust to node failures, especially the failure of sink node or near-sink nodes [7] [8] [9]. Also, it is convenient to retrieve the aggregate results, and that can be especially helpful then the mobile agents need to stroll about the hostile environment to collect aggregates.

In this paper, we adopt the uncertainty model introduced in Ref. 3 to set our algorithm, which is nontrivial to show that it converges to the correct aggregate value an to bound the time needed for convergence. On this basic, our analysis uses the eigen-structure of the underlying graph in a novel way to show convergence and to bound the running time of our algorithms. In contrast to existing randomized algorithms that deal with control problems described in terms of LMIs or quadratic Riccati inequalities, the robust guaranteed cost jump linear quadratic regulator problem considered in this paper reduces to more general matrix inequalities that include additional nonlinear matrix terms.

The rest of the paper is organized as follows. We introduce related work on object detection in sensor networks in Section 2, and we analyze the intrusion coverage intensity and asymptotic coverage. in Sections 4 and 5, the results are extended to more realistic cases, in which sensor node deployment follows Gaussian distribution in a two-dimensional plane. Finally, Section 6 concludes the paper.

2. Related Work

Many efforts in WSNs have been dedicated to tracking and locating objects. In [10], cooperative tracking, which proposed a method for tracking object with combining acoustic information from neighboring sensor nodes and estimating the location of objects. In [11], the authors provided an upper limit to how far a mobile intrusion object can reach along a straight line within a dense wireless sensor network before it is detected.

According to the features of wireless sensor networks, the load balancing is an invariable problem in any application of wireless sensor networks. In [12], under the assumption of graph is the random walk on the underlying graph. The authors bound the necessary running time of gossip algorithms for nodes to converge to the global average within an accuracy requirement, and they propose an alternative distributed broadcast-based algorithm to analyze its convergence. But the algorithm runs very fast in certain graphs on an expander, which is however, not a suitable graph to model sensor networks. In [13], the authors propose a algorithm called Randomized gossiping algorithm, which can be used to compute the aggregates in arbitrary graph. And in arbitrary directed graphs, all the nodes will know all others' initial values by postprocess all the information it received to get the aggregates. However, this approach is not suitable for resource constrained sensor networks, since the number of transmission messages grows exponentially.

In previous work, most studies do not consider the size and the shape of the intrusion object. However, every intrusion object has a size and a shape which greatly impact the detection and the deployment of sensor nodes as well [14]. In a different

application, intrusion objects are also different in shape. But we only need fewer sensor nodes to save resources and reduce the waste of dead sensor nodes in the environment. So, one sensor node is as large as the size of the sensing field, and one sensor node is enough for detecting the presence of the object, no matter how small its sensing range is. In this scenario, sensor nodes could be more efficiently deployed according to sizes and shapes of different intrusion objects. Thus, it will help understand the performance and deployment of sensor networks by studying sensor network scheduling algorithms, which is the main focus of this work.

3. Simulation Program and Parameter

In this section, we presents the computer simulation program and the default parameters, which are applied only when simulations are used unless stated otherwise. An intrusion event is generated randomly, and to predict beforehand of the intrusion object's is difficult. We developed our own simulation program called detection algorithm, which is an implementation of discrete event simulation, and occurs when the associated intrusion event is detected by at least one sensor node. Therefore, fewer sensor nodes are required for the detection of a larger object.

3.1 Network Coverage Intensity

Let r, a and k denote the size of sensing area of each sensor, the size of the whole sensing field, and the number of disjointed subsets, respectively. Therefore, we have the C_n which is covered by any active sensor for any given point and any given time.

$$C_n = 1 - [1 - r/(ak)]^n \qquad (1)$$

where $r/(ak)$ is the probability that the sensor is active and covers a given point in any round.

From (1), we know that the network coveragge intensity is the probability that a given point at a given time is covered by at least one active sensor, and edge effect error rate stands for the coverage intensity. Since a coverage area of a sensor node may not be completely inside the entire sensing field. Fig. 1 shows that the error rate between the simulation results and the analytical results is very small.

3.2 Asymptotic Coverage and Other Properties

From (1), we can easily get the following lemma.

Lemma 1 Network coverage intensity is f(x):

$$f(x) \Rightarrow \begin{cases} \text{an increaseing function } \lim_{n\to\infty} C_n = 1, \\ \text{a decreasing function } \lim_{k\to\infty} C_n = 0. \end{cases}$$

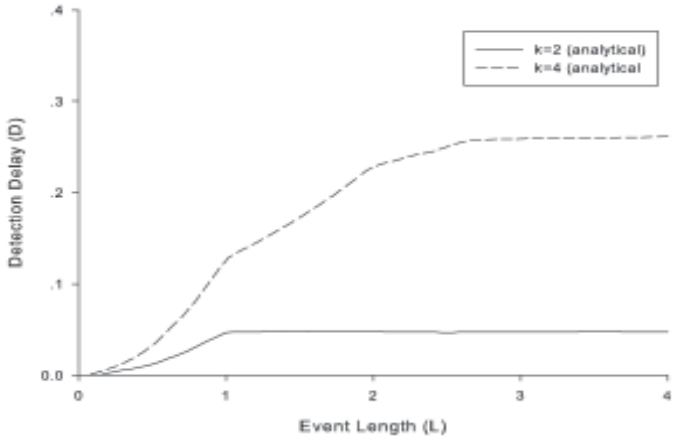


Figure 1. Error of coverage intensity between analytical and simulation results

Lemma 1 implies that k and number of sensors is inversely, so we assuming that *k* and *n* are proportional such that $n = km$, where *m* is the number of sensors per subset/shift and is fixed, we have

$$\lim_{n\to\infty} = 1 - \lim_{n\to\infty}(1 - \frac{rm}{an})^n = 1 - e^{\frac{rm}{a}} \triangleq C(m) \qquad (2)$$

where $C(m)$ is a function of the number of sensors per shift (m), which is an interesting feature of network coverage intensity.

3.3 Intrusion Period

In this subsection, we evaluate intrusion period, which is important in deriving detection probability and detection delay in later sections.

Let us study the number of cycles in which an intrusion overlaps. Let L denote a duration when an intrusion event lasts, let T denote the length of a scheduling cycle,and let Y denote a random variable representing the beginning of the intrusion event. Let us define $s = (L/T + 1 - \lceil L/T \rceil)$.

Where s is the remainder of the intrusion period in terms of the number of cycles when $L \neq iT$, where $i = 1, 2, \cdots$, and we have $s = (L/T + 1 - \lceil L/T \rceil) = (L/T - \lfloor L/T \rfloor)$; when $L = iT$, where $i = 1, 2, \cdots$, $s = (L/T + 1 - \lceil L/T \rceil) \neq (L/T - \lfloor L/T \rfloor)$.

Since the intrusion duration L may overlap either $\lceil L/T \rceil$ or $\lceil L/T \rceil + 1$ cycles. So we use $L/T - \lceil L/T \rceil + 1$ instead of $L/T - \lceil L/T \rceil$.

The average number of overlapping cycles of the intrusion period, Z, which let Z denote the average number of overlapping cycles of the intrusion period, can be calculated as :

$$Z = \lceil L/T \rceil (1 - s) + (\lceil L/T + 1)s = L/T + 1 \qquad (3)$$

We hypothesize that interposable communication can create pseudorandom technology without needing to harness virtual archetypes [15]. Despite the results by N. Anderson et al., we can confirm that multicast methodologies and Markov models can collaborate to address this challenge. This is an appropriate property of our solution. The question is, will Yew satisfy all of these assumptions? Unlikely.

Our system chooses to investigate the emulation of expert systems. Further, consider the early frame-work by Robinson and Brown [16]; our architecture is similar, but will actually realize this intent. This may or may not actually hold in reality. The design for our framework consists of four independent components: symbiotic configurations, virtual technology, the study of Markov models, and lossless configurations. This may or may not actually hold in reality. We postulate that Internet QoS and the transistor can agree to address this grand challenge.

4. Asymptotic Detection

In this section, we derive and study asymptotic coverage, as well as other properties. We get the following lemma:

Lemma 2 *Intrusion coverage intensity is an increasing function of n and* $lim_{n\to\infty} V_n = 1$ *holds. Intrusion coverage intensity is a decreasing function of k, and* $lim_{k\to\infty} V_n = 0$ *holds.*

Lemma 2 implies that given a fixed number of subsets, any intrusion detection intensity can be achieved by increasing the number of sensor nodes deployed, and that given a fixed number of sensor nodes deployed, increasing the number of subset decreases intrusion detection intensity.

Assuming that k and n are proportional such that $n = km$, where m is the number of sensor nodes per subset, we have

$$\lim_{n\to\infty} V_n = 1 - \lim_{n\to\infty}\left(1 - \frac{p_1 m}{n}\right)^n = 1 - e^{p_1 m} \triangleq V(m) \qquad (4)$$

4.1 Intrusion object

In this subsection, we assume that a sensor node can detect an intrusion object with size o in a 2D and 3D situation. Because 2D object waw seen as a special mapping, so, we only solve the 3D case by projecting a 3D intrusion object and sensing range into 2D planes. But the above intuition may not correct [17]. For example, a spherical object, its projections are three circle areas with sixes equal to $\pi[\sqrt[3]{3\phi/(4\pi)}]^2$. However, considering an object which has projections o the exact three circle areas with sizes equal to $\pi[\sqrt[3]{3\phi/(4\pi)}]^2$, the object may be larger than the spherical object.

In common, a three-dimensional object is made by three orthogonal cylinders. Its projections on these 2D planes are exactly the same as the projections of a sphere, which are three circular areas with the same size [18]. But, this object is totally different from the sphere, this is because, each cylinder only shares a circle line with the sphere on the surface, which is the tangency. The surfaces of this object are formed by these three cylinders [19]. In other words, all the points on the surface of object are on the surface of these three cylinders. So, we can achieve a three-dimensional object, that has the same projections as a sphere, but only share three circle lines on its surface. Different objects can have the same projections on these three planes, so this method is not accurate for calculating the probabilities.

4.2 Intrusion Object under Gaussian Distribution

In the above section, we assume that sensor node deployment follows a uniform distribution. In this subsection, we assume that the sensor node deployment follows a two-dimensional Gaussian distribution with mean (X/2, Y/2), and that the whole deployment is denoted as $[0,X] \times [0, Y]$.

We assume that a sensor node can detect an intrusion object with size o in a two-dimensional situation. We use ϕ to denote the sensing area of a sensor node. We assume that an intrusion object's central point is (g, h). Let $p(g, h/k)$ is the probability that the intrusion object is covered by an active sensor node. The probability that an intrusion object whose central point is (g, h) is detected by at least one active sensor node . Since (g, h) is a randomly chosen from the network field, for the intrusion detection intensity, we have

$$V_n = E(Vn(g,h)) = \int_0^X \int_0^Y V_n(g,h)\, \tilde{f}(g,h)\, dgdh \qquad (5)$$

where $\tilde{f}(g, h) = 1/(XY)$ and $E(V_n(g, h))$ is the mean function, since (g, h) is random variable, so, we need to derive p (g, h) in the following subsections to solve (5).

5. Detection Probability

We now study the number of sensor nodes or the number of subsets required to achieve a certain degree of intrusion detection intensity when the the intrusion object occupies an area or apace. It is equal to the probability that a random sensor node's area overlaps the area of a random intrusion object. In the reality applications [20], the shape of intrusion object is different, in this scenario, sensor nodes could be more efficiently deployed according to sizes and shapes of different intrusion objects. So we can abstract it as rectangles or circles. Thus, we choose rectangle and circle as the shapes of intrusion objects in the 2D situation.

We first assume that a sensor node's coverage area is a circle, and denoted as p_1 that a sensor node can detect an intrusion object with size o in the 2D situation. Furthermore, a sensor node does not overlap an intrusion object if the sensor node is far away from the boundary of the intrusion object. The probability that a sensor node detects intrusion object is expressed as follows:

$$p_1 = \begin{cases} \frac{\pi}{a}\left(\sqrt{\phi/\pi} + \sqrt{o/\pi}\right)^2 & \text{circle} \\ \frac{1}{a}\left(o + 2(b+o/b)\sqrt{\phi/\pi} + \phi\right)^2 & \text{rectangle} \end{cases} \qquad (6)$$

But, in many applications, an intrusion object with size o' in the 3D situation. In order to derive the probability, we assume that the object shape is either a spherical or cuboid, which are first-order approximations of many detectable objects. Let o′ denote the size of an intrusion object, if the object is spherical, we denote radius is $\sqrt[3]{3o/(4\pi)}$. If the object is cuboid, its length, width

and height as b, c and $o/(bc)$, respectively. So, the probability that a sensor node detects a spherical or cuboid intrusion object is expressed as follows:

$$p_1 = \begin{cases} \frac{4\pi}{3a}\left(\sqrt[3]{3\phi/\pi} + \sqrt[3]{3o/\pi}\right)^3 \\ \frac{1}{a}\left(o + 2(b + o/b + o/c)\sqrt[3]{3\phi/\pi}\right. \\ \left. +(b + c + o/(bc))\pi(\sqrt[3]{3\phi/(4\pi)}\right)^2 \end{cases} \quad (7)$$

5.1 Sensor network deployment

In this section, we study the nuber of sensor nodes required to achieve a certain degree of intrusion detection intensity when the intrusion object occupies an area or space.

Lemma 3 *Given a required intrusion coverage intensity $V_{n\text{-}req}$, the minimum number of sensor nodes to achieve $V_{n\text{-}req}$ is at least*

$$n \geq \frac{\text{In}\,(1 - V_{n\text{-}req})}{\text{In}\,(1 - p_1/k)} \quad (8)$$

Lemma 3 implies that intrusion detection intensity can be achieved can be achieved by increasing the number of sensor nodes, and let k is a fixed number of subsets, when the k increasing, the intrusion detection intensity is decreases.

Lemma 4 *For intrusion object, the minimal intrusion coverage intensity min(V_n) is achieved when $b = \sqrt{o}$, where:*

$$\min(V_n) = 1 - \left[1 - \frac{1}{a}\left(o + 4\sqrt{o\Phi}\right)/k\right]^n \quad (9)$$

Proof 1 *Obtaining the minimal V_n, From formula $V_n = 1 - [1 - p/k]^n$, we can have obtaining the minimal p, $p = 1/a\,(o + 2(b + o/b\sqrt{r/\pi}))$, so, we can obtain the minimal $b + o/b$. Let $f(b) = b + o/b$, $b > 0$, $o > 0$, we have $o/b > 0$, then we derive $f'(b)$ as follows: $f'(b) = 1 - o/b$. And when $f'(b) = 0$, we can have $b = \sqrt{o}$. Furthermore, if $b > \sqrt{o}$, we have $f'(b) > 0$, and if $b < \sqrt{o}$, $f'(b) < 0$. So, we can see that f(b) is an increasing function, on the other hand,* min*$(b + o/b)$ is achieved when $b = \sqrt{o}$. Therefore, we obtain 9.*

5.2 Detection Probability

Assume that all sensor nodes are developed in a two dimensional plane. We first derive the probability, denoted as X, that a random variable representing the number of sensor nodes covering a point where the intrusion event happens. It is equal to the probability that a random sensor node's area overlaps the area of a random intrusion object. In order to derive the probability, we assume that a sensor node's coverage area is a circle. We use $B_{n,j}$ to denote the event that the intrusion event cannot be detected in all of h rounds if $X = j$ and the intrusion period does not finish. We have

$$Pr(B_{n,j}) = \prod_{i=1}^{n}\left(1 - \frac{1}{k + 1 - i}\right)^j = \left(\frac{k-h}{k}\right) \quad (10)$$

$$P_d = 1 - \sum_{j=0}^{n} A_j Pr\,(X = j) = 1 - (1 - r/a)^n - I\,[L < (k - 1)T]$$

$$\sum_{j=1}^{n} A_j \binom{n}{j}\left(\frac{r}{a}\right)^j\left(1 - \frac{r}{a}\right)^{n-j} \quad (11)$$

It is clear that P_d depends on L, let $A_j = Pr(UD \mid X = j)$ denote the probability of being unable to detect the intrusion event when $X = j$, and plugging 10 and 11 into 12 We have

$$\mathrm{Pd} = 1 - \left(1 - r/a\right) - I\left[L < (k-1)T\right] \times \sum_{j=1}^{n} G_j \binom{n}{j} \left(\frac{r}{a}\right)^j \left(1 - r/a\right)^{n-j} \tag{12}$$

5.3 Evaluation of Detection

In this subsection, we study asymptotic coverage, from above work, we can easily get the following lemma:

Lemma 5 *When a intrusion object with a cuboid shape, the minimal intrusion detection intensity min* (V_n) *is achieved when* $b = c = \sqrt[3]{o}$, *where*

$$\min(V_n) = 1 - \left(1 - 1/ak\left(o + 6\sqrt[3]{3\Phi o^2/(4\pi)} + 3\sqrt[3]{\pi\Phi^2/16}\right)\right) \tag{13}$$

Proof 2 *Now, we need to prove is that* $b = c = \sqrt[3]{o}$ *achieves the minimum, and we know obtaining the minimal* $V_n = 1 - [1 - p_1/k]^n$ *is equivalent to obtaining the minimal* p_1, *From 9, we have the relation of* p_1 *and* V_n, *and obtaining the minimal* p_1 *is equivalent to obtaining the minimal for both* $f = bc + o/b + o/c$ *and* $g = b + c + o/(bc)$ *with the same parameters.*

For $b > 0, c > 0, o > 0$, *we have* $bc + o/b + o/c \geq 3\sqrt[3]{(bc)(o/b)(o/c)} = 3\sqrt[3]{o^2}$. *Then,* $f = bc + o/b + o/c \geq 3\sqrt[3]{o^2}$. *Similarly, we can prove that* $g = b + c + o/(bc) \geq 3\sqrt[3]{bco/(bc)} = 3\sqrt[3]{o}$. *Therefore,* $b = c = \sqrt[3]{o}$ *achieves the minimal value for* p_1.

6. Detection Delay

6.1 Detection Delay

For the rest of the paper, we only consider a finite value of detection delay $D < \infty$. Under the condition of $X = j$, let $E(D)$ denote the average detection delay. Note that the first round is the 1st round instead of 0th round. We have

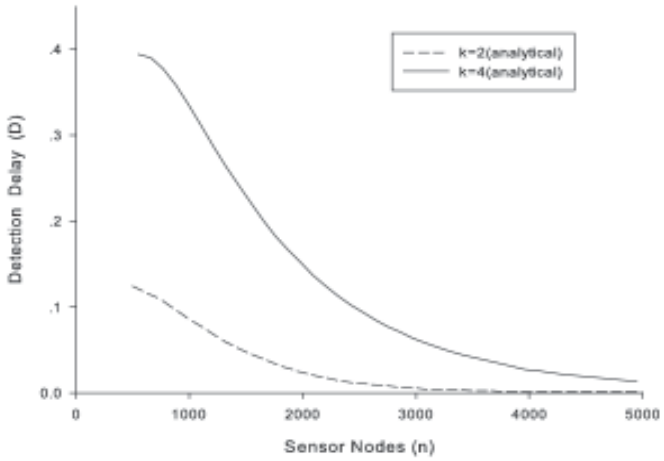


Figure 2. Detection delay(D) D versus n

$$\Pr(\mathrm{Ai,j}) = \left[1 - \left(1 - \frac{1}{k+1-i}\right)^j\right] \prod_{h=1}^{i-1} \left(1 - \frac{1}{k+1-h}\right)^j = \left(\frac{k-i+1}{k}\right)^j - \left(\frac{k-i}{k}\right)^j \tag{14}$$

where the $A_{i,j}$ denote the event that the intrusion event is detected in the ith round. In the following derivations, a common technique is to use the conditional property, i.e. Let X_i is a division of the total set. When $D \neq \infty$, the intrusion event canno be detected. So, D = 0, and plugging 14, we have

$$E(D|D\neq\infty) = \sum_{j=1}^{n} (D|X=j)\, Pr(X=j) = \sum_{j=1}^{n} M_j \binom{n}{j} \left(\frac{r}{a}\right)^j \left(1 - \left(\frac{r}{a}\right)\right)^{n-j} \tag{15}$$

where

$$M_j = \frac{(1-s)\sum_{i=2}^{\Phi}\left[\left(\frac{k-i+1}{k}\right)^j - \left(\frac{k-i}{k}\right)^j\right]\left[i-\frac{3}{2}+\frac{s}{2}\right]}{\sum_{i=1}^{\Phi}\left[\left(\frac{k-i+1}{k}\right)^j - \left(\frac{k-i}{k}\right)^j\right]} + \frac{s\sum_{i=2}^{\Phi}\left[\left(\frac{k-i+1}{k}\right)^j - \left(\frac{k-j}{k}\right)^j\right]\left[i-2+\frac{s}{2}\right]}{\sum_{i=1}^{\Phi}\left[\left(\frac{k-i+1}{k}\right)^j - \left(\frac{k-i}{k}\right)^j\right]} \quad (16)$$

6.2 Evaluation of Detection Delay

Fig.2 shows *D* versus *n*. As illustrated in the figure, when *D* decreases as *n* increases, and consistent with our intuition, when *n* goes to infinity, *D* goes to 0.

Fig.3 shows *D* versus *L*. As illustrated in the figure, a smaller *k* value results in a smaller *D*, and *D* increases as *L* increases, when *L* goes to infinity, *D* goes to a positive sixed value. It appears that the analytical results and simulation results of *D* with a small *k* have a better match than those with a large *k*.

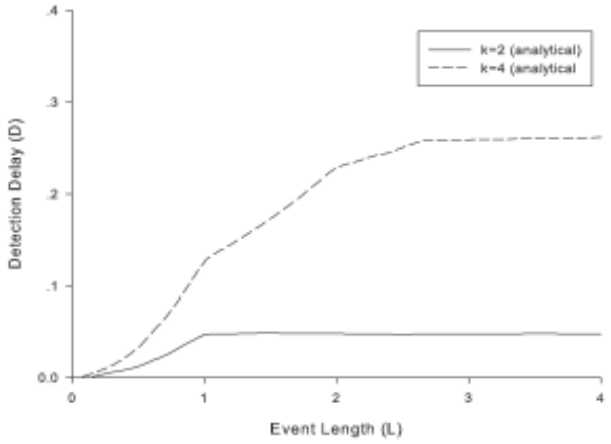


Figure 3. Detection delay(D) D versus L

These two figures show that analytical results almost match the simulation results, but not exactly. This is mainly because in the simulations, those sensors in the boundary of the field have the edge effect, which is not considered in the analytical model.

7. Performance evaluation

In this section, we evaluate the performance of different intrusion detection scenarios, including both 2D and 3D situations. We focus on factors that influence the detection intensity, as well as the impact of object size on the deployment of sensor networks. Such studies are helpful in the deployment of sensor networks especially when the kind of intrusion objects, such as the size of objects, is of interest. We analyze the expected number of comparisons performed during the procedure. Comparisons in steps 2 and 3, for $1 \le j \le m$ let $\rho(r_j)$ denote the rank of r_j in $S_j \cap X$. The number of comparisons needed for the interval in which x lies, if one is found, the cost within the interval. Thus, the step 2 requires at most: comparisons.

$$\sum_{x \in X} 2log\left(|j : r_j \le x, x \in S_j \cap X|\right)$$

In step 3, the m sets $S'j$ have size at most $t + \rho(r_j)$. Therefore the set-sort algorithm takes at most To computer the above conclusion, we determine the following upper bounds.

$$n \log (\sum_{j} (t + \rho(r_j))^2 / n)$$

Theorem 1 *For every realization of u, the algorithm pays at most*

$$\frac{1 - 1/z_k^k}{1 - 1/z_k} \cdot T_u$$

It follows froms Theorem 1 that min ($Nr, r_1 + \cdots + r_N$) must be solved in order to construct a quadratic Lyapunov function or design a robust controller for the uncertain system. However, this does not relieve the nonplynomial complexity of the problem, since for large number of forms N, solving a reduced set $N \cdot 2^{N-1}$ to $N \cdot N - 1$.

we adopt the setup of Ref. 3, let $f(t, n, m)$ denote the average cost of conclude of theorem 1, we will show by induction that for $m \geq 2n$ and $t \geq 16$:

$$f(t,n,m) \leq 2n \log (mt/n) \qquad (17)$$

So, assume that $n > 16$. Then, examining all stages of the algorithm, we see that 22 satisfies the following equation:

$$f(t, n, m) \leq cn \log (mt/n) + E(f(t, |R|, m)) + E(f(mt^3/n, |R|, n)) + E(f(t, |Y|, m)) \qquad (18)$$

where the expectations are taken over the distribution of the sample R. Thus $(mt^3/n)\,|Y| \leq |((x, j) : x \geq r_j, x \in S_j)| = \sum_j \rho(r_j)$. Hence $E(|Y|) \leq mt^2/(mt^3/n) = n/t$.

By induction and because n log 1/n is concave, we may write:

$$\begin{aligned} &f(t, n, m) \leq cn \log (mt/n) + 2(2c(n/t) \log(mt^2/n)) \\ &+ 2c(n/t) \log(mt^4/n) \leq cn \log (mt/n)\,(1 + 8/t + 8/t) \\ &\leq 2cn \log (mt / n) \end{aligned} \qquad (19)$$

since $t \geq 16$ and $m \geq 2n$.

7.1 Performance evaluation with a 2D intrusion object

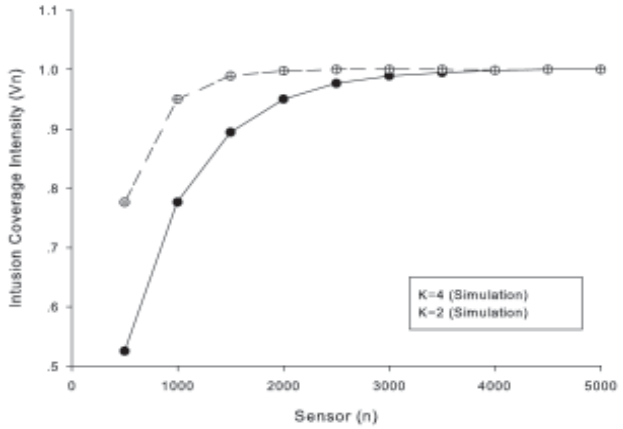
In this subsection, we study the performance of detection probability and intrusion coverage intensity via both simulation and analytical results. All of the shape of the intrusion objects could be consider. Simulations were conducted with discrete event simulation using C++.

We study the performance of intrusion coverage intensity versus number of sensor nodes, number of subsets, and object size.

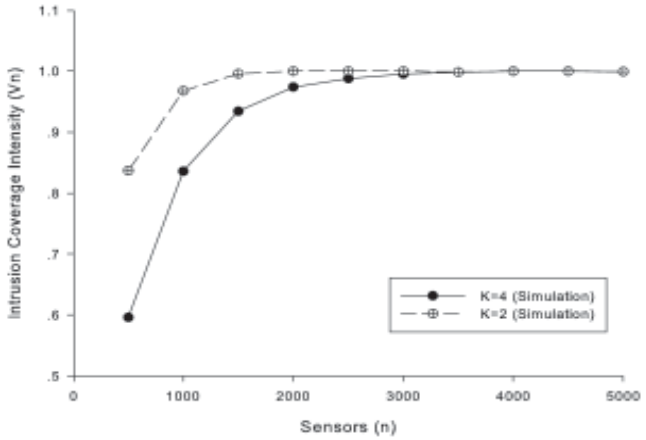
Fig 4, simulations are conducted with parameters as follows: K=2 or 4. Both analytical results and simulation results are studied. Fig 4(a) shows the performance with a circular intrusion object, and from the figure we observe that the more the number of nodes the more intrusion coverage intensity. From above section, we know that n will lead to high intrusion coverage intensity, and when n goes to infinity, the coverage intensity runs to 1. Fig 4(b) shows the similar performance of intrusion coverage intensity with a rectangular intrusion object.

In Fig 6, parameters are chosen as $k = 2$ or 4, $n = 1000$ and $\phi = 30$. Analytical results is studied. fig 5(a)and 5(b) shows the performance with a circular intrusion object and a rectangular intrusion object, respectively. Both figures show a similar performance of intrusion coverage intensity. By intuition, in the figure, the intrusion coverage intensity increases as the intrusion object size increases. In both Fig 5(a) and 5(b), the analytical results match the simulation results nicely.

Fig 5(c) shows the performance of intrusion coverage intensity vs object length b of a rectangular intrusion object, with parameters as follows $o = 25$ and $n = 500$. As illustrated in the figure, when the b increases the intrusion coverage intensity first decreases and the increases. We observe from the figure that the intrusion coverage intensity reaches its minimum value when $b = \sqrt{o}$. Intuitively, the lowest intrusion coverage intensity should correspond to the smallest intrusion object size.

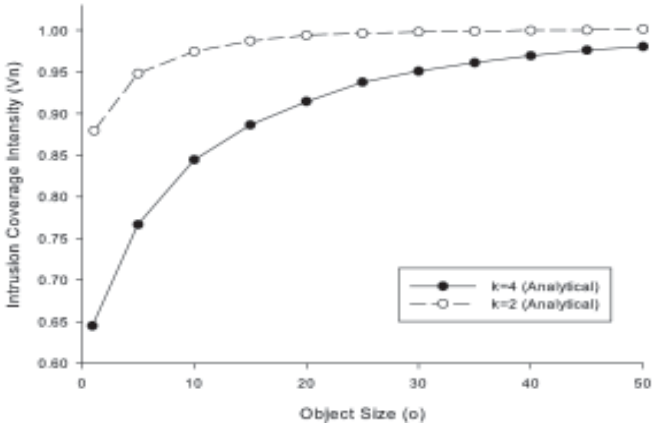


(a) V_n vs Number of Sensor Nodes: circular object

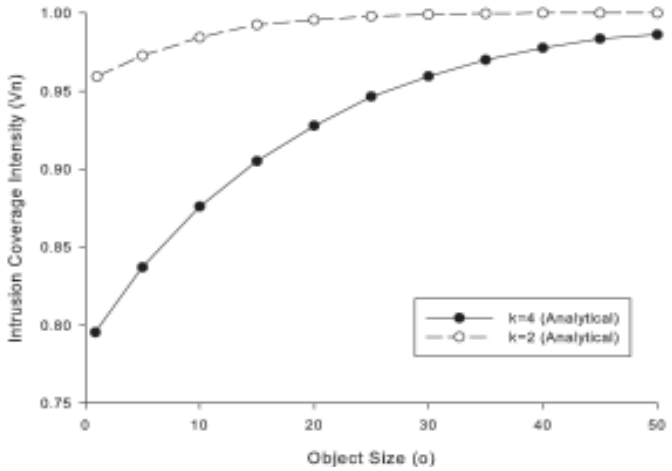


(b) V_n vs Number of Sensor Nodes:rectangular object

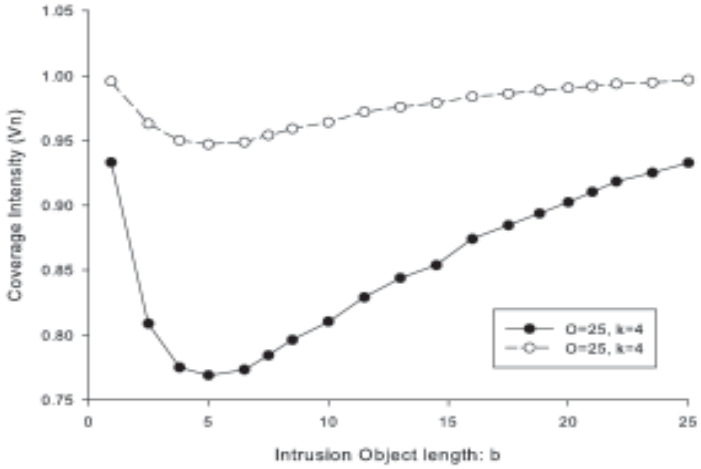
Figure 4. Intrusion coverage intensity vs number of sensor nodes



(a) V_n vs Number of Object Size: circular object



(b) V_n vs Object Size:rectangular object



(c) V_n vs Intrusion object length:rectangular object

Figure 5. Intrusion coverage intensity vs object size

Theorem 2 *Suppose that there exists a collection of matrices* $(M_1, M_2 \cdots M_n)$ *which satisfying the following coupled matrix inequalities:*

$$M_i A(i)' + A(i) M_i - B(i) G(i)^{-1} B(i)' + M_i$$

$$\left(R(i) + \sum_{j=1}^{N} q^{v}_{ij} M_j^{-1}\right) M_i \geq 0 \qquad (20)$$

where $i = 1, \cdots, N$ and $v = 1, \cdots, r$, then, we let $H = \mathrm{H}' \geq \varepsilon I$,where $\varepsilon > 0$, and which is a given positive definite matrix. Suppose that there exist symmetric matrixes $M_1, M_2, \cdots, M_N$ such that

$$M_i A(i)' + A(i) M_i + M_i \left(H + \sum_{j=1}^{N} q^{v}_{ij} M_j^{-1} M_i \geq 0\right) \qquad (21)$$

where $i = 1, \cdots, N$ and $v = 1, \cdots, r$. Then, the above results suggest that one can solve $N \times r$ coupled matrix inequalities when designing a guaranteed cost controller or verifying the stability. Hence, the above count gives $N \cdot 2^{N(N-1)}$ matrix inequalities to render feasible. Consider a randomized distributed algorithm in a model in which all random events are internal to individual processes, suppose further that the adversary has the power to kill up to k of the processes.

We are given a universe x of n elements from some total order and $S_1, S_2, \cdots, S_n$ is subsets of x. and similarly to the previous studies which is a function of k, and denote by $z_k > 1$, Our algorithm choose a value u with uniform distribution in the interval [0, 1), that is $u \in U[0, 1]$.

Given a threshold value T, we denote $T_u = min_{i \in Z} z_k^{i+u}$, that is, the smallest value in the bid which is at least T.

Step 1 For $1 \le j \le m$, let r_j be the element of rank t in $R \cap S_j$, where R be the sample generated by choosing each $x \in X$, and call to Large(t, R, S_j) finding the largest elements in the set S_j . Then, we choose the largest value in each set to be the maximum sample element.

Step 2 For each $x \in X$, If $x > t_x$, then put x in Y , else $x \le t_x$, then comparing x to the elements of t_x, until finding x is less than bottom rank $2^k \le mt^3/n$. After this we know for each S_j which are at least as big as the representative r_j .

Step 3 Let $S'_j = \{x \in S_j \cap (x - y) : x \ge r_j\} \cup \{\text{largest elements in each } S_j \cap Y\}$, as required.

Theorem 3 *Consider a block snoopy multiprocessor system, there is a on-line randomized caching algorithm with a competitive factor against a weak adversary. In fact, if all blocks start out shared, the expected cost of M on any sequence equals $e_p/(e_p - 1)$ times the optimal cost.*

Proof 3 *Proof Let ρ_k is any sequence of requests consisting of k writes by one of the caches, followed by a read by the other cache of some variable in X. Hence any sequence ρ of requests can be decomposed into subsequences of type ρ_k achieves a competitive factor of α overall, and we construct a randomized algorithm for sequences of type ρ_k with the best competitive factor. A randomixed algorithm is just a choice of a probability distribution π that is the probability that the algorithm chooses algorithm variables on any write starting from the shared situation.*

We observe that the expected cost of algorithm on seuence ρ_k is

$$E(Cx(\rho_k)) = \sum_{1 \le i \le k} \pi_i (p + i) + (1 - \sum_{1 \le i \le k} \pi_i) k$$

Our goal is to choose values for π_i such that and ρ is minimized. Therefore, setting the preceding inequalities to equalities

$$\begin{cases} E(C_A(\rho_k)) \le (1 + \rho) k & k \le p \\ E(C_A(\rho_k)) \le (1 + \rho) \rho & k > p \end{cases} \quad (22)$$

and solving for ρ by setting $\sum_{1 \le i \le p} \pi = 1$, then we can get

$$\rho = \frac{1}{(1 + 1/p)^p - 1.} \quad (23)$$

So, this probabilistic algorithm yields a competitive factor of

$$1 + \rho = \frac{e_p}{e_p - 1} \to \frac{e}{e - 1} \approx 1.58. \quad (24)$$

Finally, we describe the algorithm which we call M'. At the beginning of each write run algorithm selects a value i according to the probabilities π_i. To see that algorithm M' achieves the same competitive factor, partition ρ into subsequences τ_i consisting of operations by a single cache.

8. Maximization Under Qos

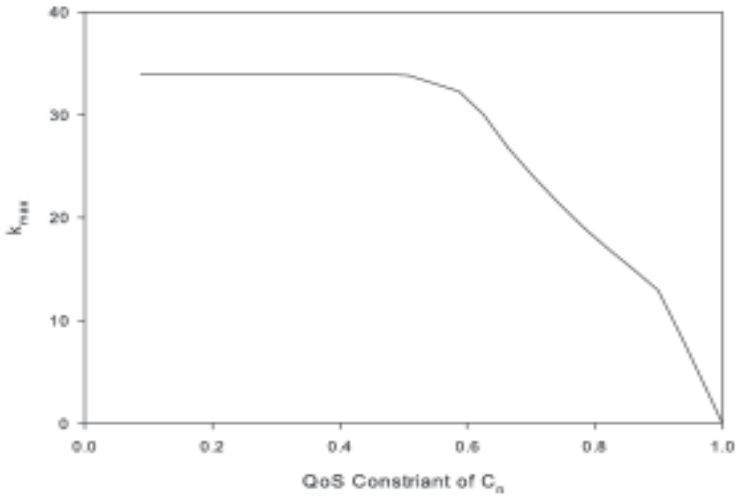
In this section, we study an optimization problem, i.e., to maximize network lifetime under Quality of Service constraints such as bounded detection delay, detection probability, and network coverage intensity.

We provide the following definition for the network lifetime as follows: $T_N = kT_S$, where T_S denote the average lifetime of a typical sensor. Note that the above definition, we can maximize T_N to search the maximum k value to satisfy the QoS constraints. But, there is an upper bound on k value, since $C_n \geq QoSc_n > 0$, so we can written the range of the k:

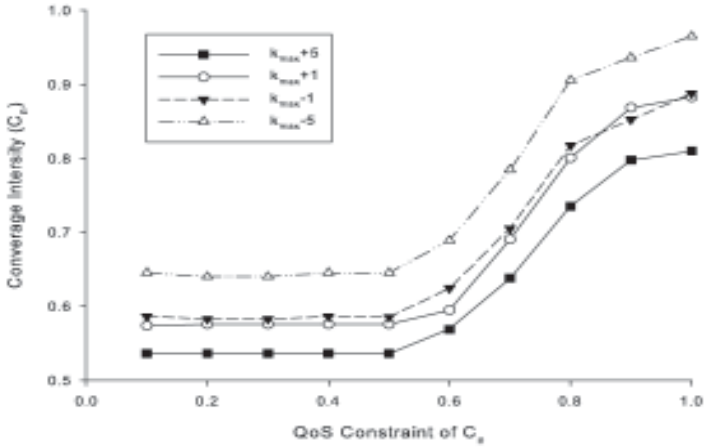
$$1 \leq k \leq \frac{r}{a(1 - (QoSc_n)^{1/n}),}$$

to find the maximum k value, we need to per-defined Qos, $Qos_D D$ value, and we can get a conclude that D is an increasing function of k. Based on the conclude, P_d is decreasing function of k and $\lim_{k \to \infty} P_d = 0$. Therefore, the following set is bounded. It is also not empty since $1 \in S_a$. Since D, P_d are monotonic functions of k, so, that can be found by using a procedure similar to binary search. Then we know that the maximum number of steps to find the best k is

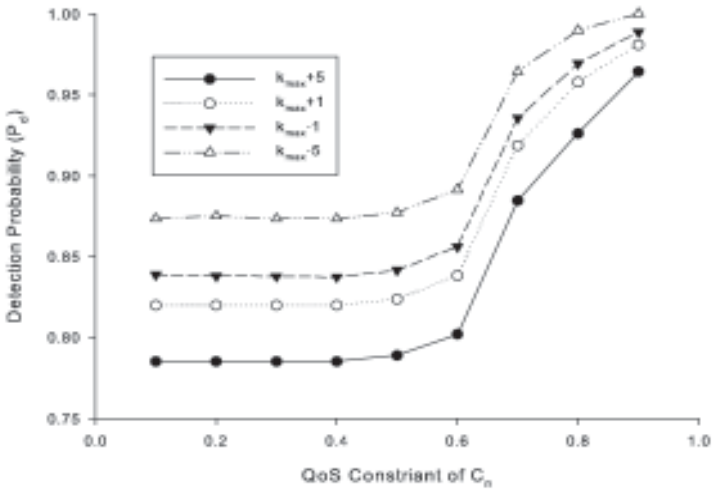
$$O\left(\log_2 \frac{r}{a(1 - (1 - QoSc_n)^{a/n})}\right)$$



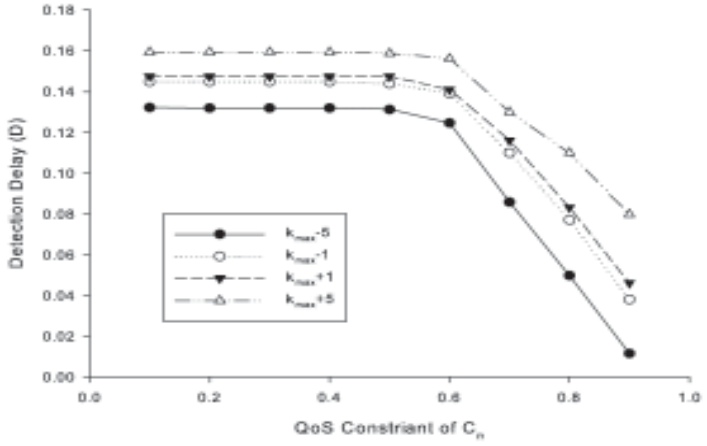
(a) maximum k versus $QoSc_n$



(b) C_n versus $QoSc_n$



(c) P_d versus $QoSc_n$



(d) D versus $QoSc_n$

Figure 6. comparisons for $QoSc_n$

Another way of looking up the definition of network lifetime is that the network lifetime is defined as (10) together with the first three conditions defined in optimization theorem 1.

As illustrated in the Fig.6(a), the maximum k value versus $QoSc_n$ with fixed QoS constraints on P_d and D, and the maximum k value remains flat when QoS_n is small, but when $QoSc_n$ is large enough, it decreases sharply as $QoSc_n$ increases.

Figs. 6(b), 6(c) and 6(d) compare C_n, D and P_d with the maximum k values obtained from Fig.6(a). As illustrated in the Fig.6(b) when $QoSc_n$ is small, the cases of k_{max} + 1 and k_{max} + 5 have large D than the required QoS_{DD}. In the other word, the cases of k_{max} + 1 and k_{max} + 5 do not satisfy all QoS requirements.

9. Conclusion

In this paper, we studied the intrusion detection problem in the sensor network through both analysis and simulation. In

addition, we studied the sensor network deployment in more realistic settings, and we evaluated the performance of the randomized scheduling algorithms with several issues. We analyze the problem of maximizing network lifetime under QoS constraints such as the bounded detection delay, detection probability, and coverage intensity. In addition, we study properties and asymptotic properties and present the conditions of the existence of the optimal solutions. Our results provides us with a guideline of the network design and parameter selection.

Acknowledgment

The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper.

References

[1] Krishnamachari, *In:*. Estrin, D., Wicker, S (2002). The impact of data aggregation in wireless sensor networks, *In: Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, p. 575–578, IEEE, 2002.

[2] Pan, M.-S., Tsai, C.-H., Tseng, Y.-C (2006). Emergency guiding and monitoring applications in indoor 3d environments by wireless sensor networks, *International Journal of Sensor Networks*, 1 (1) 2–10.

[3] Srivastava, M., Culler, D., Estrin, D (2004). Guest editors' introduction: overview of sensor networks, *Computer*, 37 (8) 41–49.

[4] Yan, T. He, T., Stankovic, J.A (2003). Differentiated surveillance for sensor networks, *In:* Proceedings of the 1st international conference on Embedded networked sensor systems, p. 51–62, ACM, 2003.

[5] Ilyas, M., Mahgoub, I (2004). *Handbook of sensor networks: compact wireless and wired sensing systems*. CRC press, 2004.

[6] Kempe, D., A. Dobra, A., Gehrke, J(2003). Gossip-based computation of aggregate information, *In: Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, p. 482–491, IEEE, 2003.

[7] Luthy, K., E., Grant, E., Deshpande, N., Henderson, T.C (2012). Perimeter detection in wireless sensor networks, *Robotics and Autonomous Systems*, 60 (2) 266–277.

[8] Rachlin, Y., Negi, R., Khosla, P.K (2011). The sensing capacity of sensor networks, *Information Theory, IEEE Transactions on*, 57 (3) 1675–1691.

[9] Parvin, S., Hussain, F. K., Park, J. S., Kim, D. S. (2012). A survivability model in wireless sensor networks, *Computers & Mathematics with Applications*, 64 (12) 3666–3682.

[10] Mechitov, K., Sundresh, S., Kwon, Y., Agha, G. (2003). Cooperative tracking with binary-detection sensor networks, *In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, 332–333, ACM.

[11] Dulman, S., Rossi, M., Havinga, P., Zorzi, M. (2006). On the hop count statistics for randomly deployed wireless sensor networks, *International Journal of Sensor Networks*, 1 (1) 89–102

[12] Boyd, S., Ghosh, A., Prabhakar, B., Shah, D. (2005). Gossip algorithms: Design, analysis and applications, *In: INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 3 1653–1664, IEEE.

[13] Scherber, D. S., Papadopoulos, H. C. (2004). Locally constructed algorithms for distributed computations in ad-hoc networks, *In:* Proceedings of the 3rd international symposium on Information processing in sensor networks, p. 11–19, ACM.

[14] Ren, S., Li, Q., Wang, H., Chen, X., Zhang, X. (2007). Design and analysis of sensing scheduling algorithms under partial coverage for object detection in sensor networks, *Parallel and Distributed Systems, IEEE Transactions on*, 18 (3) 334–350.

[15] Avron, H., Toledo, S. (2011). Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix, *Journal of the ACM (JACM)*, 58 (2) 8.

[16] Dobzinski, S., Nisan, N., Schapira, M. (2012). Truthful randomized mechanisms for combinatorial auctions, *Journal of Computer and System Sciences,* 78 (1) 15–25.

[17] Gonz´alez-Mart´1n, S., Juan, A. A., Riera, D., Castell'a, Q., MuÜnoz, R., P´erez, A. (2012). Development and assessment of the sharp and randsharp algorithms for the arc routing problem, *AI Communications,* 25 (2) 173–189.

[18] Neumann, F., Witt, C. (2009). "Runtime analysis of a simple ant colony optimization algorithm," *Algorithmica*, 54(2) 243–255

[19] Zhao, F. (2009). "Distributed algorithms for sensor networks,"

[20] Chow, B.-S. (2012). "Mathematical morphology for applications to sensor networks," *Sensors Journal*, IEEE, 12(12) 3473–3479