Reliable and Energy Efficient Target Coverage for Wireless Sensor Networks

Jing He**, Shouling Ji, Yi Pan, Yingshu Li

Department of Computer Science, Georgia State University, Atlanta, GA 30303-3044, USA

Abstract: A critical aspect of applications with Wireless Sensor Networks (WSNs) is network lifetime. Power-constrained WSNs are usable as long as they can communicate sense data to a processing node. Poor communication links and hazardous environments make the WSNs unreliable. Existing schemes assume that the state of a sensor covering targets is binary: success (covers the targets) or failure (cannot cover the targets). However, in real WSNs, a sensor covers targets with a certain probability. To improve WSNs’ reliability, we should consider that a sensor covers targets with users’ satisfied probability. To solve this problem, this paper first introduces a failure probability into the target coverage problem to improve and control the system reliability. Furthermore, we model the solution as the $\alpha$-Reliable Maximum Sensor Covers ($\alpha$-RMSC) problem and design a heuristic greedy algorithm that efficiently computes the maximal number of $\alpha$-Reliable sensor covers. To efficiently extend the WSNs lifetime with users’ pre-defined failure probability requirements, only the sensors from the current active sensor cover are responsible for monitoring all targets, while all other sensors are in a low-energy sleep mode. Simulation results validate the performance of this algorithm, in which users can precisely control the system reliability without sacrificing much energy consumption.

Key words: target coverage; wireless sensor networks; energy efficiency; sensor scheduling; $\alpha$-reliable maximum sensor covers; node failure

Introduction

Wireless Sensor Networks (WSNs) consist of a large number of ad-hoc networked, low-power, short-lived and unreliable micro-sensors, which are limited in computation, memory capacity and radio range\textsuperscript{[1]}. Many sensors are deployed in regions of interest to collect related information or report that some event has taken place in that area. Therefore, WSNs are widely applied to battlefield surveillance, health care applications, environment and habitat monitoring, home appliances, smart spaces, and inventory tracking\textsuperscript{[2]}. A basic and important function of WSNs is to monitor areas or targets for a long period, such as battlefields. Since sensors are often deployed in remote or inaccessible environments where replenishing the sensor energy is usually impossible, a critical issue in WSN applications is conserving sensor energy and prolonging the network lifetime while guaranteeing the coverage of desired areas or targets, which is called the Coverage problem.

The coverage problem is a fundamental problem in WSNs for environment monitoring and surveillance purposes. The coverage concept is subject to a wide range of interpretations due to the variety of sensors and applications. Generally, coverage which has direct effect on the network performance can be considered as the measure of quality of service in a WSN\textsuperscript{[3]}. In
general, coverage problems either study deployed sensors to cover the sensing field completely\cite{4,5}, or make sure that all the sensing field is covered by a certain amount of sensors, such as 1-coverage or \(k\)-coverage\cite{6,7}, or select active sensors in a densely deployed WSN to cover all the sensing field\cite{8-12}. The last one is known as an Activity Scheduling Problem (ASP)\cite{13}. Recently, a lot of researchers have paid attention to ASP. Generally speaking, according to the requirement of sensing tasks, ASP may be divided into four classes: target coverage\cite{9,12}, area coverage\cite{8,14}, barrier coverage\cite{10}, and patrol coverage\cite{11}.

In this paper, we address an ASP for target coverage. As is already known, the goal of ASP is to maximize the network lifetime on the premise of preserving the sensing coverage. Many algorithms propose to organize sensors into a number of subsets, such that each set completely covers all the sensing field, and schedules times to make these subsets activated successively, in which only one set is active at any time instant. By avoiding redundant sensors wasting their energy, the network lifetime is prolonged. The above problem is NP-hard\cite{9}, so many optimization techniques are applied, such as genetic algorithms, linear programming, greedy algorithms\cite{15-17}, etc. However, another prominent issue in the target coverage problem is how to improve reliability of the whole system. To our best knowledge, there are few works on this aspect. Nevertheless, due to interference of dangerous environments, nodes may become unavailable (e.g., physical damage, lack of power resources), malfunction or totally missing. Improving reliability of target coverage in WSNs is worth more research effort in the future. In this work, we also consider the reliability of the target coverage problem.

Our approach differs from the aforementioned solutions by adding reliability restrictions to get non-disjoint sensor sets and by allowing the sets to operate for different time intervals. Specifically, our approach is to find the active non-disjoint sensor sets to fully cover all targets in a distributed manner to maximize the network lifetime within a preset reliability threshold. We introduce a new concept which is the failure probability of sensor sets for reliability. Only when the failure probability of sensor sets is greater than the preset threshold, can these sets be set to active. The reliability of a network is a measure of the Quality of Service (QoS) of the sensing function and is subject to a wide range of interpretations due to a large variety of sensors and applications. Thus in this study, we use the failure probability to be the metric. All the current works use at least one node to cover targets for energy saving. However, one critical node’s failure might cause the failure of the whole network. Hence we choose the lowest failure probability sensor set to monitor all targets, which will improve the network lifetime as well as the reliability of the whole system.

Consequently, our main contributions in this work are summarized as follows.

- We introduce the failure probability concept into the target coverage problem. To our best knowledge, this is the first study addressing the problem in the literature. Almost all the related works make the assumption that if the sensor is not out of energy, it can cover the targets in its sensing range. However, in reality, there is a failure probability associated with each sensor as well as sensor sets.

- We formalize the target coverage problem with reliability as a \(\alpha\)-Reliable Maximum Sensor Cover (\(\alpha\)-RMSC) problem.

- We propose a distributed algorithm to solve the \(\alpha\)-RMSC problem. The algorithm can precisely control the failure rate of the whole system, which is a critical fact in many applications, such as military surveillance systems, and environment monitoring systems.

- The simulation results validate that the proposed algorithm has perfect reliability control and also has similar network lifetime compared to the latest algorithm\cite{9}. Thus the proposed algorithm does not sacrifice the energy consumption to achieve system reliability.

The rest of the paper is organized as follows. In Section 1, we review previous related works. In Section 2, we first introduce some preliminaries, such as network models and related definitions, and then formalize the \(\alpha\)-RMSC problem. In Section 3, a heuristic greedy algorithm is proposed to solve the \(\alpha\)-RMSC problem. We show the simulation results in Section 4. Finally we conclude this paper and give future work directions in Section 5.

1 Related Work

The coverage problem in WSNs has been intensively investigated. In this section, we summarize the related
studies according to two groups: the target coverage problem and other coverage problems.

1.1 Target coverage

There exists a wealth of literature on the target coverage problem in WSNs. We can divide the target coverage problem into (1) target coverage problem with unique sensing range\(^{[9,18]}\), (2) target coverage problem with multiple sensing ranges\(^{[17,19]}\), (3) target coverage problem in directional WSNs\(^{[20]}\), and (4) other variants, such as target area coverage\(^{[13]}\), in which the sensors cover not only a set of targets, but also deal with a small area coverage, and target \(Q\)-coverage\(^{[21]}\), in which each target needs to be covered by different numbers of sensors.

The authors in Refs. \([9,18]\) propose an efficient way to extend the network lifetime by organizing the sensors into a maximal number of sensor sets that are activated successively. Only the sensors from the current active set are responsible for monitoring all targets and for transmitting the collected data, while all other nodes are in a low-energy sleep mode. The authors in Ref. \([17]\) consider the status scheduling of a sensor with multiple sensing ranges. A collaborative task scheduling algorithm is proposed to minimize the energy consumption. It employs a two-level scheduling approach to the execution of tasks collaboratively at a group and individual levels among neighboring sensor nodes. The authors in Ref. \([19]\) propose an Integer Linear Programming (ILP) based approximation to maximize the network lifetime, where the sensors have different sensing ranges. In Ref. \([20]\), the authors organize the directions of sensors into a group of non-disjoint cover sets. One cover set which can cover all the targets satisfying their coverage quality requirement is activated at one time. In Ref. \([13]\), the authors propose geometric based activity scheduling scheme to address the target area problem. By means of computational geometry, the sensors can self-determine when to sleep or wake up while preserving the coverage requirement. In Ref. \([21]\), the authors prove that the target \(Q\)-coverage problem is NP-Hard. A greedy algorithm is proposed to efficiently solve the target \(Q\)-coverage problem.

1.2 Other coverage

Many studies have focused on characterizing the area coverage and designing algorithms to achieve desired area coverage. In Ref. \([22]\), the authors study the coverage of a grid-based WSNs. They derive the necessary and sufficient conditions on the sensing range and failure rate of sensors in order to ensure that the whole network is covered as well as connected. In Ref. \([23]\), a probing-based density control algorithm was proposed to extend the network lifetime. Again, the basic idea is to turn off redundant sensors to save energy. A sleeping sensor wakes up occasionally to probe its neighborhood and starts working if there are no other working sensors in its probing range. The desired redundancy of working sensors can be achieved by adjusting the probing range of sensors.

Another line of work on coverage studies the path exposure of moving objects in WSNs, which is a quantitative measure of how well sensors can detect objects moving in the network. In Ref. \([24]\), the authors propose algorithms to find paths which are most or least likely to be detected by sensors in a WSN. The authors further define and study the path exposure of a moving object in a WSN\(^{[25]}\), which is a quantitative measure of how well an object, moving on an arbitrary path, can be detected by a WSN. An algorithm is developed to find minimum exposure paths in WSNs, where the probability of a moving object being detected is minimized. Along this line, Ref. \([26]\) investigates deployment strategies for WSNs performing target detection. The goal of sensor deployment is to maximize the exposure of the least exposed path in the network.

Different criterias can categorize the current coverage problem into different types. One type includes partial coverage and complete coverage problem. In Ref. \([27]\), a Fractional Coverage Scheme (FCS) was proposed to achieve the desired partially coverage with a minimum number of active sensors for energy conservation. Another one includes connected coverage and non-connected coverage problem. In Ref. \([28]\), the authors proposed a density control algorithm for WSNs to keep as few as possible sensors in active state to achieve a connected coverage of a specific area of interest.

1.3 Remarks

Most of these previous works focused on saving energy to prolong network lifetime. In this paper, we
approach the coverage problems from a different perspective by studying the reliability of the whole system. The reliability of the whole system is determined by the basic network parameters (e.g., deployment strategy, sensor density, physical environment, etc.), however it fundamentally impacts the performance in a WSN. For example, one critical node’s failure might cause the failure of the whole network. Hence, controlling and improving reliability of the target coverage problem in WSNs is worth more research effort.

## 2 Network Model and Related Definitions

Before solving the target coverage problem, we give some basic definitions and notations used throughout this paper.

### 2.1 Network model

We consider a WSN consisting of a set $S$ of $n$ sensors that are also called nodes. Each $s \in S$ can sense $m$ interested targets (denoted by $T$) in its sensing range and communicate with nodes in its transmission range. Each target $t$ has a unique identification number denoted by $t_j$. We make the natural assumption that there do not exist two sensors at the same location. Also, each sensor $s \in S$ has a unique identification number, denoted by $s_i$. The sensors are distributed over a large 2-dimensional area. We refer to the region as the monitoring area.

We assume that the sensing and transmission ranges of node $s$ are open discs, centered at $s_i$, with radial $r_s$ and $R_s$ respectively, where $R_s > r_s$. We assume a sensor $s_i$ covers a target $t_j$ if the Euclidean distance between the sensor $s_i$ and the target $t_j$ is smaller or equal with the sensing range $r_s$. Nodes are described as adjacent or neighbors if they are included in the transmission range of each other.

We assume each sensor $s_i$ has a Failure Probability associated with each target $t_j$, in the monitored area (denoted by $f_{p_{ij}}$). The failure probability is initially given by the manufacturer. However there are lots of factors that affect the parameter, such as the weather in the monitoring area, interference to the sensors, or unexpected accidents. If we know the environment in advance, we can predict or calculate the Failure Probability easily.

The time is divided into time slots and we assume that the sensors have synchronized clocks which notify them at the beginning of each time slot. Sensor $s_i \in S$ has an initial energy $B_i$ and, as a normalization, we assume that every sensor consumes $e$ units of energy in each time slot in which it is active. For saving energy, a sensor may be in a sleep mode, in which it does not communicate with its neighbors nor sense its vicinity. A sensor in the sleep mode consumes only negligible unit energy, which is assumed to be zero. This is also called duty-cycling WSNs.

### 2.2 Related definitions

All targets in the monitoring area are covered by at least one sensor which is called the target coverage problem in WSN. In most cases, sensors are densely deployed to tolerate failures. It is not reasonable to turn on all sensors in the monitoring area to cover all the targets, because more than one sensor can cover the same target. So it is necessary to divide the $n$ sensors into a couple of subsets, and each subset can cover all interested targets. In each time slot, only one subset is active. The duty to monitor targets is cycled in the whole WSNs. The main purpose of the duty-cycling is to save energy and prolong the lifetime of the WSN.

**Definition 1** Target Set. The target set is the set of all targets in the monitoring area, which is denoted by $T = \{t_1, t_2, \ldots, t_n\}$.

**Definition 2** Sensor Cover. Given a WSN consisting of a set of $n$ sensors, $S = \{s_1, s_2, \ldots, s_m\}$. Any subset $C$, of $S$ that can completely cover the target set is termed as a sensor cover, $C = \{s, s_2, \ldots, s_k\}$, if we can find a maximum $k$ such sensor covers, then $r \in [1, k]$.

If there does not exist a sensor cover $C$, such that all the nodes in $C$ have non-zero energy, then the network is said to have a coverage hole.

**Definition 3** Network Lifetime. The network lifetime is the time interval from the activation of the network till the first time at which a coverage hole appears.

Taking the reliability into consideration, the definition of the classic target coverage problem needs to be improved. In reality, the failure of sensors cannot be ignored, so we add the Failure Probability as a property to each sensor. Then we can calculate the Failure Probability for each sensor cover.
Definition 4 Target Failure Probability. Given a target $t_j$ in the Target Set $T = \{t_1, t_2, \ldots, t_m\}$, we can find the sensor set $S_j = \{s_1, s_2, \ldots, s_l\}$, $\forall j \in [1, m]$, where $S_j$ is a subset of the whole sensor set $S$. Every node $s_i$ in the sensor set $S_j$ can cover the given target $t_j$ with the probability $f_{p_{ij}}$. The Target Failure Probability (denoted by $tp_{j}$) is defined as the probability that sensor $s_i$ fails to cover target $t_j$.

$$tp_{j} = \prod_{i=1}^{l} f_{p_{ij}}$$ (1)

As we have already mentioned there is more than one sensor that can cover the same target. So we can divide the whole sensor set into some non-disjoint sensor covers. Let some sensor covers be active and some sensor covers sleep, then the network lifetime can be prolonged. In this situation, how to calculate the target failure probability becomes complex. Let us assume there are $l$, where $l \in [1, n]$, sensors that can cover target $t_j$. Each sensor $s_j$ has a failure probability $f_{p_{ij}}$ to cover the target $t_j$. So the failure probability to cover the target by all $l$ sensors should be the product of each sensor’s failure probability, i.e.,

$$fp_{ij} * fp_{i2} * \cdots * fp_{il},$$

which is formulated in Eq. (1). The purpose of calculating the Target Failure Probability is to calculate the sensor cover Failure Probability. Next we introduce the definition of Sensor Cover Failure Probability.

Definition 5 Sensor Cover Failure Probability. The probability that a sensor cover $C_{c} = \{s_1, s_2, \ldots, s_{k}\}$, $\forall i \in [1, n]$; $\forall r \in [1, k]$, where $k$ is the maximum number of sensor covers we can find, failing to cover all the targets in the Target Set $T = \{t_1, t_2, \ldots, t_m\}$, which is denoted by $cp_{c}$.

$$cp_{c} = 1 - \prod_{j=1}^{m} (1 - tp_{j})$$ (2)

$$= 1 - \prod_{j=1}^{m} (1 - l \sum_{i=1}^{l} f_{p_{ij}})$$ (3)

This definition is used to monitor the reliability of each sensor cover. The higher the sensor cover Failure Probability is, the more reliable the sensor cover is. Hence when we schedule the sensor covers, we know how to choose the desired sensor cover to be active to improve the whole network’s reliability.

From Definition 2, we know every sensor cover must completely cover all targets in the monitoring area. Based on Eq. (1), we can calculate each target’s failure probability $tp_{j}$, where $j \in [1, m]$. The failure probability is equal to $(1 - \text{Success Probability})$. So we first calculate the Success Probability of covering each target, which is $1 - tp_{j}$. Then the Success Probability of covering the target set $T = \{t_1, t_2, \ldots, t_m\}$ is $(1 - tp_{j}) * (1 - tp_{j_2}) * \cdots * (1 - tp_{j_m})$. By using Eq. (1), we deduct the above Success Probability of covering the target set, and get Eq. (2). If we apply Eq. (1) to Eq. (2), then we get Eq. (3).

Figure 1 shows an example with four sensor nodes $S = \{s_1, s_2, s_3, s_4\}$ and three targets $T = \{t_1, t_2, t_3\}$. The coverage relations between sensors and targets are also illustrated in Fig. 1. From the coverage information we can find all 4 sensor covers, i.e., $C_1 = \{s_1, s_2, s_3\}$, $C_2 = \{s_2, s_3\}$, $C_3 = \{s_2, s_4\}$, and $C_4 = \{s_3, s_4\}$, shown in Fig. 2. For convenience, we assume 1 is the life time of each sensor. Therefore, if we let set $C_1$ be active for a 0.33 time slot, $C_2$ active for a 0.5 time slot, $C_3$ active for a 0.33 time slot, and $C_4$ active for 0.33 time slot, then the network lifetime will be 1.49 time slots. Thus, duty-cycling WSN can improve the network lifetime.

Equation (1) is used to calculate each target’s failure probability. Let us use target 1 ($t_1$) as an example. From the coverage information in Fig. 1, we know that there are three nodes $S_1 = \{s_1, s_2, s_3\}$, which can cover $t_1$, so the Target Failure Probability of $t_1$ is

$$tp_1 = f_{p11} \ast f_{p21} \ast f_{p31}$$ (4)

Fig. 1 Example with three targets and four sensors

Fig. 2 Illustration of four sensor covers
Based on Eq. (2), we can calculate each sensor cover’s failure probability. If we use sensor cover $C_3 = \{s_2, s_3\}$ as an example, then

$$\text{cp}_3 = 1 - \prod_{j=1}^{3} (1 - \text{tp}_j) = 1 - (1 - \text{tp}_1)(1 - \text{tp}_2)(1 - \text{tp}_3) = 1 - (1 - \text{fp}_{p2})(1 - \text{fp}_{p2} \times \text{fp}_{p2})(1 - \text{fp}_{p2})$$  \hspace{1cm} (5)

Previous algorithms did not consider the network reliability. Hence, the network lifetime of our scheme cannot be compared directly to the lifetime of previous algorithms. Thus, we proposed a new measurement matrix:

**Definition 6** Reliable Lifetime. Reliable Lifetime is equal to the current active sensor cover’s failure probability times the current network lifetime.

### 2.3 Problem formulation

In this subsection, we proceed to define the $\alpha$-RMSC problem.

**Definition 7** $\alpha$-RMSC Problem. Given a collection $S$ and a finite set $T$, we find a family of sensor covers $C$ with time weights $t_{w_1}, t_{w_2}, \ldots, t_{w_i}$ in $[0, 1]$ and sensor cover Failure Probabilities $\text{cp}_1, \ldots, \text{cp}_k$, where $k$ is the maximum number of sensor covers we can find, such that

1. $\forall i \in [1, k], \text{cp}_i \leq \alpha$, where $\alpha$ is the user pre-defined maximum failure probability.
2. Maximize $\sum_{j=1}^{i} t_{w_j}$, where each sensor $s$ appears in $C$ with a total weight of at most 1, where 1 is the life time of each sensor.

From Definition 7, we know that we want to determine a number of sensor covers $C_1, C_2, \ldots, C_i$, where each sensor cover $C_i, r \in [1, k]$ completely covers all the targets with $\text{cp}_j \leq \alpha$, such that to maximize the network lifetime $\sum_{j=1}^{t} t_{w_j}$, where $\forall i \in [1, k], t_{w_i}$ is the time interval while the sensor cover $C_i$ is active. Note that if a sensor belongs to more than one sensor cover, then the sum of the time intervals of those sensor covers cannot outnumber 1. This is because each sensor cannot be active more than its lifetime 1.

The solutions of the $\alpha$-RMSC problem guarantee that the success probability of covering $T$ is larger than $1 - \alpha$. So by setting different values of $\alpha$, WSNs gain different reliable levels of solutions.

### 3 Our Proposed Algorithm

We now describe the $\alpha$-RMSC heuristic algorithm. The $\alpha$-RMSC problem is closely related to the MSC problem, which is recently proposed to solve the traditional target coverage problem in Ref. [9]. The difference is that we are trying to maximize the network’s reliability as well as the network lifetime. The MSC problem is proven to be a NP-Hard problem. From the work in Ref. [9], we can claim that the $\alpha$-RMSC problem is also NP-Hard. Hence, we provide a heuristic approach to solve the $\alpha$-RMSC problem.

#### 3.1 $\alpha$-RMSC heuristic algorithm overview

Our heuristic algorithm (see Algorithm 1) takes $S$ (Sensor Set), $T$ (Target Set), $e$ (Sensor Lifetime Granularity, where $e \in (0, 1)$), and $\alpha$ (the pre-set threshold) as the input parameters. The algorithm returns all possible sensor covers $C_1, C_2, \ldots, C_k$.

**Algorithm 1** $\alpha$-RMSC Heuristic Algorithm

**Require** $S$: Sensor Set; $T$: Target Set; $e$: Sensor Lifetime Granularity; $\alpha$: The Maximum Failure Probability;

**Ensure** $C_1, \ldots, C_k$: $\alpha$-Reliable sensor covers

1: **for** each $s$ in $S$ **do**
2: \hspace{1cm} $\text{s.lifetime} = 1$;
3: **end for**
4: $\text{SENSORS} = S$;
5: $i = 0$;
6: **while** each target is covered by at least one sensor in $\text{SENSORS}$, and the $\text{SENSORS.cp} \leq \alpha$ **do**
7: \hspace{1cm} $i = i + 1$;
8: \hspace{1cm} $C_i = \emptyset$;
9: \hspace{1cm} **while** $C_i.cp \leq \alpha$ **do**
10: \hspace{2cm} $s = \text{Greatest_Contribution_Sensor}(T, S, C_i)$;
11: \hspace{2cm} $C_i = C_i \cap s$;
12: **end while**
13: **for** each sensor $s \in C_i$ **do**
14: \hspace{1cm} $\text{s.lifetime} = \text{s.lifetime} - e$;
15: \hspace{1cm} **if** $\text{s.lifetime} = 0$ **then**
16: \hspace{2cm} $\text{SENSORS} = \text{SENSORS} - s$;
17: **end if**
18: **end for**
19: **end while**
20: **return** $C_1, \ldots, C_i$,
Algorithm 1 recursively builds sensor covers from line 6 to 19. The set SENSORS maintains the list of sensors that have residual energy greater than zero, thus these sensors participate in additional sensor covers. At each step, a target is selected in line 6 to be covered. Once the target has been selected, the algorithm selects the sensor with the greatest contribution that covers the selected target. Sensor contribution functions (i.e., Greatest_Contribution_Sensor function shown in Line 10 in Algorithm 1) are defined in Algorithm 2. Once a sensor has been selected, it is added to the current sensor cover in line 11. When all targets are covered, the new sensor cover is formed.

Algorithm 2 Greatest_Contribution_Sensor

Require \( T; S; C_r \);
Ensure candidate_sensor; greatest contributed sensor;

1: greatest_contribution = 0;
2: for each sensor \( s \in (S - C_r) \) do
3: contribution = 0;
4: for each target \( t_j \in T \) do
5: if \( s \) covers \( t_j \) then
6: current_tp = 1;
7: for each sensor \( cs_i \in C_r \) do
8: if \( cs_i \) covers \( t_j \) then
9: current_tp = current_tp * cs.fp_j;
10: end if
11: end for
12: contribution = contribution + current_tp * (1 - cs.fp_j) * w_j;
13: end if
14: end for
15: if contribution > greatest_contribution then
16: greatest_contribution = contribution;
17: candidate_sensor = s;
18: end if
19: end for
20: return candidate_sensor.

3.2 Contribution function

How to find the greatest sensor contribution is critical to solve the \( \alpha \)-RMSC problem. The contribution of a sensor \( s_j \) to a sensor cover \( C_r \) is defined as follows:

**Definition 8** Contribution. We assume sensor \( s_j \) can cover a set of targets \( \{t_1, t_2, \ldots, t_l\}, l \in [1, m] \), the contribution of \( s_j \) to \( C_r \) is

\[
\text{Con}(C_r, s_j) = \sum_{i=1}^{k} \text{tp}_j \ast (1 - \text{fp}_p) \ast w_i
\]  

The assignment of weight \( w_1, w_2, \ldots, w_i \) is based on the application specification. Generally, we can give evenly distributed weight values. However in some cases, targets are mostly sparsely covered, then the weight value for this target should be a large number.

We still use Fig. 1 as the example to illustrate the contribution function. If Algorithm 1 puts \( s_1 \) and \( s_2 \) into the current sensor cover. The algorithm cannot stop at this point, because the target \( t_1 \) has not been covered yet. The next step is to decide which sensor needs to be put into the current sensor cover, \( s_3 \) or \( s_4 \). At this time we need to calculate the contribution of \( s_1 \) and \( s_4 \) respectively according to Eq. (6). From Fig. 1, we know that \( s_1 \) can cover \( \{t_1, t_2\} \) and \( s_4 \) can cover \( \{t_3, t_4\} \). To simplify the calculation, we assume all sensors’ Failure Probabilities are the same, denoted by \( p \). Then,

\[
\text{Con}(\{s_1, s_2\}, s_1) = \text{tp}_1 \ast (1 - \text{fp}_p) \ast w_1 + \text{tp}_2 \ast (1 - \text{fp}_p) \ast w_2 = p^2 \ast (1 - p) + 1 \ast (1 - p) = (p^2 + 1)(1 - p)
\]

(7)

\[
\text{Con}(\{s_1, s_2\}, s_4) = \text{tp}_2 \ast (1 - \text{fp}_p) \ast w_3 + \text{tp}_1 \ast (1 - \text{fp}_p) \ast w_4 = p \ast (1 - p) + 1 \ast (1 - p) = (p + 1)(1 - p)
\]

(8)

From the result of Eqs. (7) and (8), we know Con(\{s_1, s_2\}, s_1) < Con(\{s_1, s_2\}, s_4), which means that \( s_4 \) contributes more to sensor cover \( \{s_1, s_2\} \) than \( s_1 \) does. Thus, \( s_4 \) will be added to the current sensor cover. The result is reasonable intuitively. \( \{s_1, s_2\} \) covers \( t_1 \) twice and \( t_2 \) once. Both \( s_1 \) and \( s_4 \) cover \( t_1 \). However, \( s_3 \) can only cover \( t_2 \), which is only covered once by \( \{s_1, s_2\} \). Hence, \( s_4 \) might improve the reliability to cover \( t_2 \) compared with \( s_1 \).

The Algorithm 2 recursively calculate each target’s Target Failure Probability currentTp from line 7 to 11. The contribution (see details in Definition 8) of each unchosen sensor to the current sensor cover is calculated based on Eq. (6) in line 12. The most contributed sensor candidate_sensor is outputted in line 20.

3.3 Relations between MSC and \( \alpha \)-RMSC

Each solution of our proposed \( \alpha \)-RMSC is also a solution of MSC\[^9\]. Unlike the situation in MSC, where only the lifetime of sensors is taken into consideration,
in $\alpha$-RMSC, we also use the failure probability for each sensor cover as the selection criterion, which may be especially useful in the applications that are sensitive to failures. An important point to be indicated is that a lower failure probability means a higher network reliability and might lead a shorter network lifetime. Therefore we need a tradeoff based on actual needs. Moreover, when we set $\alpha$ as 1, the $\alpha$-RMSC problem converts to the MSC problem.

4 Performance Evaluation

In this section we evaluate the performance of $\alpha$-RMSC and MSC\(^9\). We simulate a stationary network with $n = 36$ sensor nodes (each with an initial energy 1 and each time slot consumes $e$ energy), deployed uniformly at random in a $30 \times 20$ m area. $m = 3$ target points are randomly put in the monitoring area. We assume the sensing range, transmission range, and failure probability are equal for all the sensors in the network. We examine the lifetime attained by $\alpha$-RMSC compared to the lifetime of MSC. Each time slot is 1 unit long. In the simulation, we consider the following tunable parameters.

- $fp$, the failure probability of each sensor node. We vary the number from 0.02 to 0.88 with an increment of 0.02.
- $\alpha$, the user pre-defined maximum failure rate of the whole network. We vary the number from 0.015625 to 1 multiplying by 2 each time.
- $e$, Sensor Lifetime Granularity. We vary the number from 0.0125 to 1 multiplying by 2 each time.

4.1 Simulation 1: Control failure probability

In this simulation, we show how well our algorithm controls the system’s failure probability to satisfy different user pre-defined $\alpha$ values.

In Fig. 3, we measure the reliable lifetime (see details in Definition 6) and the failure probability when the preset maximum failure probability $\alpha$ varies from 0.015625 to 1 using our $\alpha$-RMSC algorithm. From Figs. 3b and 3c, we find the failure probability of the network is controlled very well. The circle-dotted line is completely under the rectangle-dotted line. The rectangle-dotted line is the user required maximum failure probability. The circle-dotted line is the failure probability calculated by Eq. (3). Based on our proposed $\alpha$-RMSC algorithm, as the number of iteration increases, the failure probability must be less than or equal to the preset maximum failure probability $\alpha$. The simulation results verify our proposed algorithm. In Fig. 3a, since $\alpha = 0.015625$, which is a small number, the limitation of the number of sensors deployed in a fixed area causes the convergence difficulties of the algorithm. Thus, the network reliable lifetime is also unstable. However, in Fig. 3b ($\alpha = 0.5$), the algorithm
converges quickly, roughly at iteration 20. The network reliable lifetime decreases when the iteration number increases as we use a greedy criterion to find \( \alpha \)-Reliable sensor covers. In Fig. 3c (\( \alpha = 1 \)), it converges at iteration 20. There is a trade-off between the network lifetime and the network reliability. A smaller \( \alpha \) gives a higher network reliability, but a smaller network lifetime. When \( \alpha \) is set to 1, the \( \alpha \)-RMSC problem converts to the MSC problem.

4.2 Simulation 2: Comparison between \( \alpha \)-RMSC and MSC

In this section, we first analyze the performance of MSC\(^9\), and then compare the performance of MSC to \( \alpha \)-RMSC the scheme.

From Fig. 4, we measure the network reliable lifetime and failure probability when \( e \) varies from 0.125 to 1 using the MSC algorithm. The network reliable lifetime decreases as the sensor lifetime granularity \( e \) increases. If in each time slot, every sensor consumes more energy, then the network lifetime should decrease. The network reliable lifetime increases with the number of iterations, as each node can now participate in more sensor covers and then the overall failure probability increases as well. As a conclusion from Fig. 4, \( e \) values do not change the network reliability much, however, they do affect the network lifetime.

In Fig. 5, we plot the reliable lifetime computed by \( \alpha \)-RMSC and MSC\(^9\) depending on the number of iterations. We vary the user pre-set threshold value \( \alpha \) from \( \frac{1}{2^5} \) to 1, multiplying by 2 each time. We only show \( \alpha = 0.5 \) and \( \alpha = 1 \) in Fig. 5. From the figure, we can see when \( \alpha = 1 \), the reliable lifetime of \( \alpha \)-RMSC is close to the lifetime of MSC. This is reasonable, because our \( \alpha \)-RMSC algorithm spent some energy to calculate the sensor cover’s failure probability. As \( \alpha \) becomes smaller, the object is to improve the network reliability rather than to improve the network lifetime. Thus, we found the lifetime line goes down quickly in Fig. 5, where \( \alpha = 0.5 \).

The simulation results can be summarized as follows:

- for a specific number of targets, the network lifetime output by our \( \alpha \)-RMSC increases with the number of sensors;
for a specific number of sensors, the network lifetime increases as the number of targets to be monitored decreases;

• sensor lifetime granulation $e$ only affects the network lifetime. It does not affect the network reliability much;

• for smaller maximum failure probabilities $\alpha$, the lifetime decreases over time as less sensor covers can be found;

• There is a trade-off between a higher lifetime and the reliability of the whole system;

• $\alpha$-RMSC has a fast converge time, thus it is scalable to large sensor networks.

5 Conclusions and Future Work

WSNs are battery powered, therefore prolonging the network lifetime through a power aware node organization is highly desirable. Poor communication links and hazardous unknown environments make it necessary to improve reliability of WSNs. An efficient method for saving energy and controlling system reliability is to schedule the sensor node activity such that every sensor alternates between sleep and active state to meet the user preset maximum failure probability. One solution is to organize the sensor nodes in sensor covers, such that every cover completely monitors all the targets. These covers are activated in turn, such that at a specific time only one sensor set is responsible for sensing the targets, while all other sensors are in sleep state. The failure probability of each sensor cover must be greater than the user preset maximum failure probability. This problem is modeled as the $\alpha$-Reliable Maximum Sensor Covers problem, which is a NP-Hard problem. Moreover, we propose an efficient heuristic greedy algorithm to solve the problem. Simulation results demonstrate that our proposed method can control the system’s reliability easily without sacrificing the network lifetime much.

In the future, we will investigate the relations between sensor density and the maximum failure probability. If the maximum failure probability is given by users, then we can calculate the sensor density of the whole network. Therefore, we do not need to deploy a large number of sensors to improve the system’s reliability. Another direction is to design a distributed algorithm for the energy-efficient reliable target coverage problem.

References


