Deployment Policies to Reliably Maintain and Maximize Expected Coverage in a Wireless Sensor Network

Nicholas T. Boardman
University of Arkansas, Fayetteville

Deployment Policies to Reliably Maintain and Maximize Expected Coverage
in a Wireless Sensor Network

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Engineering

by

Nicholas T. Boardman
United States Air Force Academy
Bachelor of Science in Economics, 2014
Air Force Institute of Technology
Master of Science in Operations Research, 2016

July 2021
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

Kelly M. Sullivan, Ph.D.
Dissertation Director

Edward Pohl, Ph.D.
Committee Member

Hugh Medal, Ph.D.
Committee Member

Shengfan Zhang, Ph.D.
Committee Member

Abstract

The long-term operation of a wireless sensor network (WSN) requires the deployment of new sensors over time to restore any loss in network coverage and communication ability resulting from sensor failures. Over the course of several deployment actions it is important to consider the cost of maintaining the WSN in addition to any desired performance measures such as coverage, connectivity, or reliability. The resulting problem formulation is approached first through a time-based deployment model in which the network is restored to a fixed size at periodic time intervals. The network destruction spectrum (D-spectrum) has been introduced to estimate reliability and is more commonly applied to a static network, rather than a dynamic network where new sensors are deployed over time. We discuss how the D-spectrum can be incorporated to estimate reliability of a time-based deployment policy and the features that allow a wide range of deployment policies to be evaluated in an efficient manner. We next focus on a myopic condition-based deployment model where the network is observed at periodic time intervals and a fixed budget is available to deploy new sensors with each observation. With a limited budget available the model must address the complexity present in a dynamic network size in addition to a dynamic network topology, and the dependence of network reliability on the deployment action. We discuss how the D-spectrum can be applied to the myopic condition-based deployment problem, illustrating the value of the D-spectrum in a variety of maintenance settings beyond the traditional static network reliability problem. From the insight of the time-based and myopic condition-based deployment models, we present a Markov decision process (MDP) model for the condition-based deployment problem that captures the benefit of an action beyond the current time period.

Methodology related to approximate dynamic programming (ADP) and approximate value iteration algorithms is presented to search for high quality deployment policies. In addition to the time-based and myopic condition-based deployment models, the MDP model is one of the few addressing the repeated deployment of new sensors as well as an emphasis on network reliability. For each model we discuss the relevant problem formulation, methodology to estimate network reliability, and demonstrate the performance in a range of test instances, comparing to alternative policies or models as appropriate. We conclude with a stochastic optimization model focused on a slightly different objective to maximize expected coverage with uncertainty in where a sensor lands in the network. We discuss a heuristic solution method that seeks to determine an optimal deployment of sensors, present results for a wide range of network sizes and explore the impact of sensor failures on both the model formulation and resulting deployment policy.

Contents

List of Tables

List of Figures

List of Published Papers

- N. T. Boardman and K. M. Sullivan, "Time-based node deployment policies for reliable wireless sensor networks," *IEEE Transactions on Reliability*, pp. 1-14, 2021. [Accepted][Chapter 2]
- N. T. Boardman and K. M. Sullivan, "Condition-based node deployment policies for reliable wireless sensor networks," *Proceedings of the 2021 IISE Annual Conference.* [Accepted with minor revisions][Chapter 3]
- N. T. Boardman and K. M. Sullivan, "Approximate dynamic programming for condition-based node deployment in a wireless sensor network." [Submitted][Chapter 4]

Disclaimer

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

1 Introduction

The performance of a wireless sensor network (WSN) is heavily influenced by the capabilities of individual sensor nodes and their locations throughout the network. Sensors must contribute towards a network communication objective and be located near other sensors to transmit and receive messages. Sensors must also be sufficiently scattered throughout the region of interest to ensure a high degree of coverage is achieved. Network coverage and connectivity are two important measures when describing the current condition of the WSN, and are also impacted over time as a result of sensor failures. Individual sensor nodes contain a finite power supply supporting node operation such as monitoring the surrounding area and communicating with neighbor sensor nodes. As an increasing number of sensor nodes deplete their power supply and no longer contribute towards WSN functions, the status of the overall network begins to deteriorate as well.

To maintain the WSN over a prolonged period of time it is necessary to repair, replace, or deploy new sensors in the network. Repairing or replacing sensors require an ability to directly access the network and are further complicated by the environment the WSN is deployed in, as well as the size of the network with respect to both the area monitored and the number of sensors. WSNs are typically characterized by the low cost of individual sensor nodes and the lack of infrastructure (e.g., cables, wires, etc.) supporting network operations, both of which contribute towards the deployment of new sensors as an attractive option. Additionally, sensors are not required to be placed deterministically in the WSN; they can be randomly deployed over a region in sufficient number to provide a high degree of coverage and connectivity.

In this work we discuss several different models to maintain a WSN through the deployment of new sensors in the network. Whereas previous research on sensor deployment related problems has focused on a network coverage or connectivity measure, our work focuses on evaluating deployment policies with respect to reliability. Chapter 2 presents a time-based deployment model and is defined by a deployment interval δ and the number of sensors in the network n_1. The main contribution is a model that balances cost and reliability, where reliability is estimated using the network destruction spectrum (D-spectrum). The significance of this model is that it decouples the complexity of the two decision variables where the D-spectrum is influenced by n_1 but independent of δ. We also present an efficient destruction algorithm which performs a vital subroutine in estimating the D-spectrum. The improved destruction algorithm allows for a larger number of replications, and reduces the variance in the resulting reliability estimate.

In Chapter 3 we discuss a myopic condition-based deployment model that is defined by an inspection interval δ and mission budget β. The model allows more control over how new sensors are deployed by selecting a smaller subregion of the network for a sensor to be randomly located in. As a result WSN reliability is complicated further by the decision on the number of new sensors deployed in each subregion. We discuss how the D-spectrum can be applied to approximate the reliability for a given deployment policy, avoiding the computational effort of a traditional Monte Carlo reliability method. The resulting problem formulation is also significant as the model can now be viewed as a decision on how a fixed number of sensors are allocated to different subregions in the network. While determining an optimal deployment policy remains a difficult task due to the dependence on network structure and unknown relationship between different deployment policies, a number of policies anticipated to result in a highly reliable network can now be estimated and compared in an efficient manner. Building on Chapter 2, Chapter 3 further explores the value of the D-spectrum in a maintenance model and the ability to estimate reliability as new sensors are deployed in the WSN.

Chapter 4 introduces a Markov decision process (MDP) model for the condition-based deployment problem to incorporate the impact of sensor deployment on WSN performance beyond the next δ time units as well. The model is defined by an inspection interval δ and starting budget B_0 which is available to deploy new sensors over a series of missions. With greater control on how the budget is utilized the model places a larger emphasis on the number of sensors deployed each mission, and more importantly deciding when the network is in a state that does not require maintenance and the budget can be preserved for future time periods. To determine an optimal deployment policy we discuss approximate dynamic programming (ADP) methodology based on a combination of state aggregation functions and a lookup table value function approximation. An approximate value iteration algorithm is applied to solve for an optimal deployment policy and we explore the impact of both the inspection interval and starting budget on the performance of the resulting policy. The benefit of the ADP policy is also demonstrated by comparing the performance to an appropriate myopic condition-based and time-based deployment policy.

Chapter 5 approaches a sensor deployment problem with an expected coverage objective instead of the previous reliability based objectives. We present a stochastic optimization model that determines an optimal deployment of sensors in various subregions throughout the network, with uncertainty in the exact placement of a sensor and an objective of maximizing the resulting WSN coverage. To solve for an optimal deployment policy we discuss a scenario based approach to sample the location for every sensor in the network. The model requires a large number of scenarios to properly reflect the randomness in sensor location and obtain an accurate estimate on the expected coverage of a deployment policy. Initial results indicate that an exact approach is intractable for even a small number of scenarios and motivates a heuristic solution method. We present a heuristic approach combining elements of greedy search, neighborhood search, and bread-first search algorithms, and discuss the results on a range of test instances. Finally, we discuss a simple extension of the model to address a probability of sensor failure and explore the impact on the resulting deployment policy.

2 Time-Based Deployment Policies for Reliable Wireless Sensor Networks

Abstract

Wireless Sensor Networks (WSNs) are commonly used to monitor a remote environment over an extended period of time. One important design consideration is the WSN's reliability of area coverage, as sensors fail over time and functionality of the network degrades. When the WSN no longer sufficiently covers the region, maintenance actions may consider repairing failed nodes or deploying new sensors to reestablish network capability. Towards identifying an optimal maintenance policy, specifically the deployment of new sensors, we present an optimization model formulated using the network destruction spectrum (D-spectrum), that seeks to determine a time-based deployment policy balancing cost and reliability. While the benefits of using the D-spectrum in reliability are widely researched, the application of the D-spectrum to enable the modeling and solving of an optimization problem is new. With the complexity already present in estimating reliability, the significance of this optimization model is that it decouples the complexity of estimating the D-spectrum from the estimation of network reliability in the presence of a given deployment policy. This key feature allows us to quickly evaluate a wide range of time-based deployment policies. Additionally, we present an efficient destruction algorithm that performs a vital subroutine in estimating the D-spectrum, allowing for a larger of number of replications to be performed in the Monte Carlo simulation thereby reducing the variance of the resulting reliability estimate. Finally, the optimization model is illustrated through a numerical example.

2.1 Introduction & Literature Review

Wireless sensor networks (WSNs) consist of a set of sensors, distributed over a region of interest, that monitor and report desired conditions within the region. The number of sensors in these networks can vary greatly depending on the coverage required, the detection and communication capabilities of sensors, as well as the initial effort to design and allocate sensors across the network. The network application can further influence network size ranging from areas such as fire and flood detection [1], military operations with battlefield tracking and surveillance, or environmental control in buildings [2]. Another attractive feature of WSNs is that they can be designed and constructed for a specific application (such as those previously mentioned), but also offer the flexibility to be quickly deployed as required. Whether a network is specially designed or randomly deployed can be impacted by the application or operational setting. For example, in areas with harsh environmental conditions or rough terrain, sensors can be air dropped over a desired location to achieve coverage in a given area [3]. A consequence of this approach is that sensors are randomly deployed throughout the region, but the lack of control over specifically locating sensors can be offset by deploying a larger number of sensors. The low-cost characteristic of sensors is an additional component that contributes towards the random deployment of a larger number of sensors as a feasible strategy [4].

Once established it is important that the network operate for a sufficient period of time, particularly when sensors are deployed in remote areas and difficult to access for repair. The performance of a WSN is primarily impacted by the number of operating sensors and the ability of these sensors to communicate with each other [5]. These are both characteristics of the network that decline over time as sensors start to fail. The lifetime of an individual sensor is bounded by a battery or power supply, and once diminished the sensor no longer operates [6]. Sensors additionally have components required for monitoring, processing, and routing information through the network. Software errors in any of these functions, potentially in the form of failing to properly send/receive information from nearby sensors, result in a drop in network capability and may propagate failures through the network [7]. Hardware failures also arise with physical damage and can possibly cause components to break, particularly when operating in a harsh environment where sensors are exposed to weather or other external factors [2].

The failure process has led to research on methods to extend sensor lifetimes, commonly through topology control algorithms. By modifying the communication range the power consumed by a sensor can be managed while still ensuring a message can be routed through the network [8]. Energy consumption can further be controlled by specifying which paths are used to route data, as well as aggregating data to avoid sending duplicate messages [9]. Similarly, the network topology can be dynamically controlled through periodically turning sensors on and off. Such a sleep/wake schedule results in redundant nodes conserving energy until required to help prolong network lifetime [6, 10]. These algorithms commonly aim to maximize the time until the first sensor fails, but sensor networks often have redundancy built in and can tolerate some sensor failures without losing capability [11]. Another limitation is that sensor lifetime is treated to be bounded by a battery supply that is consumed at some known rate and once depleted the sensor fails. Addressing random sensor failures that can arise (e.g., environmental interference, physical damage [4]) adds a layer of difficulty to estimating network lifetime, particularly when we are interested in the status of the network beyond the first sensor failure.

Another approach to extend network lifetime is through the use of movement based connectivity methods. With sensor nodes randomly deployed throughout the region it may be desirable to relocate sensors immediately after deployment in an effort to improve the overall network coverage and connectivity [12, 13]. We may also be interested in re-positioning sensor nodes in response to failures that occur [14–16]. One of the advantages of a movement based approach is that network topology can be dynamically controlled to prolong network lifetime. However the cost of mobile sensors is typically significantly larger than static sensors [4, 15] which introduces questions about their suitability for a large scale WSN of interest.

The attractiveness of topology control algorithms and movement based connectivity methods is their ability to extend the lifetime of a given network. The long-term operation of a WSN must also consider deploying new sensors in the network, particularly in the presence of an increasing number of sensor failures. In [17], different node replacement policies are examined to maintain a coverage requirement to maximize lifetime where a decision to replace a failed sensor or not is made immediately after observing a failure, however only a small number of sensors can be replaced. A similar problem focuses on deploying new sensors to restore some level of connectivity and/or coverage, with the additional challenge of deploying the fewest number of new sensors [18–20]. Problems related to optimal node placement commonly fall in the NP-Hard class of problems [21], which motivates the search for approximation algorithms. One of the primary limitations of current models is that they are framed in the context of single stage. That is, the deployment of new sensors is concerned with immediately preserving network functionality, but does not consider the future failure probability of sensors. As a result it is reasonable to question the reliability of the network after sensors have been deployed. To the best of our knowledge, attempts at creating a durable network with the deployment of additional sensors focus on providing a level of redundancy or k-connectivity [20, 22]. This is certainly a desirable characteristic for the network, but a node redeployment strategy should also be influenced by the residual life distribution of sensors and how frequently such a policy needs to be implemented.

Existing research has focused primarily on extending network lifetime (common in topology control and movement based methods) or maintaining a coverage/connectivity requirement (common in node redeployment methods), but there appears to be a lack of emphasis on analyzing a maintenance policy with respect to network reliability. The focus of this work is directly concentrated on evaluating and comparing the performance of time-based maintenance policies, specifically the deployment of new sensors in the network, with respect to both cost and estimated reliability. One of the difficulties of this task is that network reliability problems commonly fall in the #P-Complete class of problems [23] and can be difficult to solve exactly, particularly for larger size networks or when reliability estimation is performed as a subroutine in another algorithm. As a result network reliability problems are routinely solved by approximate solution methods.

One theme that arises with respect to approximate methods is to bound network reliability. Compared to an exact method that explores every possible network state, carefully selecting a subset of states to evaluate can lead to more efficient algorithms that provide upper and/or lower bounds on network reliability [24]. Depending on the manner in which bounds are constructed, these algorithms still require a large amount of effort particularly for larger sized networks [25].

Closely related, and often utilized within bounding techniques, is to use a Monte Carlo method to estimate network lifetime. A naive/crude Monte Carlo approach is to randomly generate a failure time for each sensor according to its life distribution, order the sensor failures, and then examine the network state after each successive sensor failure to determine the instant of network failure. Repeating this process allows for an estimation of overall network reliability upon completion. One main drawback of this approach is the unbounded growth of the relative error for highly reliable and highly unreliable networks [26]. This issue has been addressed through the use of improved Monte Carlo methods [27] and variance reduction techniques [25, 28].

A crude Monte Carlo method can also be improved leveraging the destruction spectrum (D-spectrum) of the network, also referred to as the network signature, where the resulting reliability estimation has bounded relative error [29]. Under the assumption of independent and identically distributed sensor lifetimes, the destruction spectrum also yields an efficient representation of the network's reliability but depends only on the system structure [30]. While both the D-spectrum and crude Monte Carlo approaches require solving an embedded destruction problem in order to determine the time at which all sensors are disconnected from one or more sink nodes, we show the destruction problem for the network signature can be solved more efficiently than the one for crude Monte Carlo.

With an understanding of network reliability we can now begin to explore the impact from the deployment of new sensors in the network. The objective of deploying new sensors is directed at restoring network function (i.e., as a corrective maintenance action) or improving network capability (i.e., as a preventive maintenance action) [31]. Depending on the application of the WSN, a temporary failure of the network may lead to serious consequences making corrective deployment policies unattractive. For this reason we focus on preventive deployment policies which could be based on the number of functioning sensors, the size of the region the network covers, or the time since the last deployment action. Preventive policies have also been explored in related network maintenance models, such as power distribution networks studied in [32, 33]. Also discussed is the added difficulty in that we must now estimate the cost of such an action as well to compare different policies, in addition to estimating network reliability in the presence of a deployment policy.

Time-based (or periodic) deployment is one version of a preventive deployment policy in which new sensors are deployed at fixed time intervals. We examine a time-based deployment policy in a network consisting of n_1 sensors, where every δ time units new sensors are deployed in the network to increase the number of functioning sensors back up to n_1. An alternative action is to repair a failed sensor node, but given the low-cost characteristic of sensors combined with the potential difficulty in accessing a specific sensor for repair, the deployment of new sensor nodes is an attractive policy. Additionally, WSNs lack the requirement for a physical connection between sensors (e.g., wire, cable, etc.) which further avoids the need to repair failed sensors.

Towards identifying an optimal time-based deployment policy, the main contribution of this work is an optimization model, formulated using the D-spectrum, that seeks to determine a time-based deployment policy balancing cost and reliability. While the benefits of using the D-spectrum in reliability are widely researched, the application of the D-spectrum to enable the modeling and solving of an optimization problem is new. The inclusion of the D-spectrum in determining an optimal time-based deployment policy is of interest as the D-spectrum is a property of the network structure impacted by n_1, which can be viewed as a network design variable, but is independent of the deployment interval δ. As a result the Monte

Carlo simulation for estimating the signature, which may be a large source of computational effort, is not impacted by the time-based deployment policy. This modeling approach thus decouples the complexity of estimating the network signature for evaluating reliability, and estimating reliability in the presence of a given time-based deployment policy. In a related effort, we demonstrate how to exploit random geometric graph structure commonly used to model WSNs to efficiently update the destruction spectrum estimate for networks of varying size, yielding further computational advantages in the optimization model.

Additionally, we provide an efficient destruction algorithm that performs a vital subroutine in estimating the D-spectrum and in turn reliability for a WSN. This algorithm is based on recognizing that a single iteration of a Monte Carlo algorithm (for estimating the D-spectrum) yields a maximum capacity path problem. The D-spectrum's unique characteristics enable Dial's implementation of Dijkstra's algorithm to be utilized when solving for the maximum capacity path. This improvement in the algorithm allows for a larger number of replications, thereby reducing the variance of our reliability estimate beyond the traditional approach of estimating the D-spectrum. We also discuss a simple extension that allows network reliability to be calculated for different coverage requirements without increasing the complexity of the algorithm.

The remainder of this work is organized as follows. Section 2.2 summarizes the modeling of the WSN and the methodology to estimate reliability in the network both with and without the deployment of new sensors. Section 2.3 presents a destruction algorithm necessary for estimating the network signature, which is then used to estimate reliability and evaluate various time-based deployment policies. Section 2.4 conveys the procedure of optimizing time-based deployment policies of a WSN, which is then demonstrated in Section 2.5.

Section 2.6 summarizes conclusions and directions for further work.

2.2 Modeling Fundamentals and Network Reliability

We model a WSN as a network whose node set consists of a sink node 0, sensor nodes $\{1, \ldots, n_1\}$, and target nodes $\{n_1 + 1, \ldots, n_1 + n_2\}$. We define $\mathcal{N}_1 = \{0, 1, \ldots, n_1\}$ as the set of sink and sensor nodes and $\mathcal{N}_2 = \{n_1 + 1, \ldots, n_1 + n_2\}$ as the set of target nodes. The two main functions of a sensor node are to communicate with other sensor nodes and to monitor targets. Sensor nodes $i \in \mathcal{N}_1$ and $j \in \mathcal{N}_1$ are capable of communicating with one another provided they are within a given range $d_1 > 0$ of each other. Let $\mathcal{E} \subseteq \binom{\mathcal{N}_1}{2}$ denote the set of undirected edges $\{i, j\}$ created due to each pair of sensor nodes $i \in \mathcal{N}_1$ and $j \in \mathcal{N}_1$ that can communicate, and define $\mathcal{A}_1 = \bigcup_{\{i,j\} \in \mathcal{E}} \{(i, j), (j, i)\}$ as the expanded, directed edge set associated with $\mathcal{E}$. A sensor node is capable of monitoring any target within a range $d_2 > 0$. Let $\mathcal{A}_2 \subseteq \mathcal{N}_1 \times \mathcal{N}_2$ denote the set of directed edges that defines which targets are covered by which sensors. Thus, an arc $(i, j) \in \mathcal{A}_2$ exists if sensor $i \in \mathcal{N}_1 \backslash \{0\}$ monitors target $j \in \mathcal{N}_2$. Without loss of generality, going forward we assume a single sink node is located in the network. We can always transform the network to one that contains a single sink by adding a new artificial sink node, and adding an arc from this new node to every sensor connected to one of the original sink nodes. The original set of sink nodes and their adjacent arcs are then removed.

In what follows, it will be useful to consider both the directed network $\mathcal{G} = (\mathcal{N}_1 \cup \mathcal{N}_2, \mathcal{A}_1 \cup \mathcal{A}_2)$ and the undirected network $\mathcal{G}' = (\mathcal{N}_1, \mathcal{E})$ as representations of the WSN. For brevity, we define $\mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2$ and $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$. An example of both networks for $n_1 = 100$ sensor nodes randomly distributed over a $[0, 1] \times [0, 1]$ region is illustrated in Figure 1.

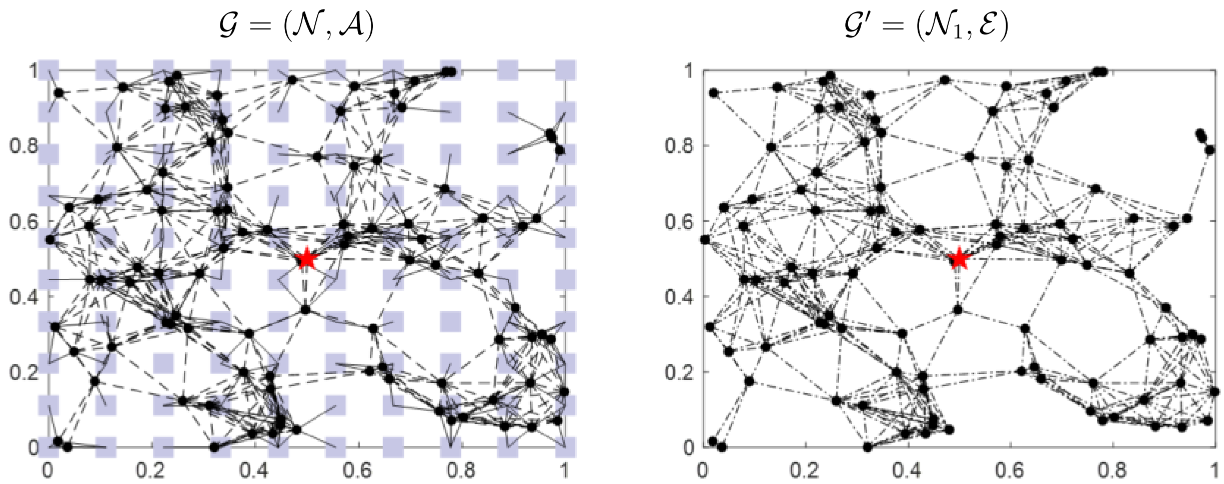

Figure 1: Example realization of the network $\mathcal{G}$ and $\mathcal{G}'$ over the $[0,1] \times [0,1]$ region, $n = 100$, $d_1 = 0.2$, $d_2 = 0.1$, and a single sink node (marked by "$\star$") located at $(0.5, 0.5)$. The set of 10×10 target nodes marked by "■" is defined as $\mathcal{N}_2 = \{0.00, 0.11, 0.22, \ldots, 1.00\} \times \{0.00, 0.11, 0.22, \ldots, 1.00\}$. Note that for ease of illustration in the network $\mathcal{G}$, the edge pair (i, j) and (j, i) is represented by a single dashed arc, while a solid arc represents a sensor to target arc.

Due to the failure of sensor nodes the WSN evolves over time. At any time $t \geq 0$ the network $\mathcal{G}$, so that $\mathcal{G}'(t)$ is well defined, is represented by $\mathcal{G}(t)$, and consists of only sensors that are still functioning, indicated by the set $\mathcal{N}_1(t)$, and the resulting communication edges (i.e., excludes failed nodes from $\mathcal{N}_1$ and adjacent edges). Let $T \geq 0$ represent the lifetime of a sensor, $F(t)$ denote the cumulative distribution function (cdf) of T, and $\bar{F}(t) = 1 - F(t)$ the survival function of T. Upon generating a deterministic failure time for each sensor from the distribution of T, let π represent the order of sensor failures where $\pi(k) = i$ if node $i \in \mathcal{N}_1 \backslash \{0\}$ is the k^{th} sensor to fail, and let $q_i \in \{1, \ldots, n_1\}, i \in \mathcal{N}_1 \backslash \{0\}$, be the index such that $\pi(q_i) = i$. For the network constructed in this manner, the following assumptions are also imposed.

Assumption 1 *Sensor lifetimes are independent and identically distributed (i.i.d.).*

Assumption 2 *Sensor capabilities are identical.*

Under Assumption 1, each sensor has the same life distribution T and sensors fail independently of one another. This yields favorable theoretical properties with respect to the D-spectrum we can leverage to develop more efficient algorithms. We later discuss the impacts of relaxing the assumption that lifetimes are identically distributed, while maintaining independent failures, in the context of deploying new sensors in the network. Assumption 2 implies that all sensors also have the same communication radius d_1 and common sensing radius d_2. With identical sensors, this alleviates concerns of sensor compatibility and integrating multiple sensor types to function together.

Assumption 3 *The sink node is perfectly reliable.*

For a target to be covered it must be both within the coverage radius of a functioning sensor, and there must exist a communication path from this monitoring sensor back to the sink. Given this requirement, it is clear that the sink node is one of the most important nodes in the network. We can guarantee that if the sink node fails we no longer cover any of the targets, and the network fails as well. For this reason, with Assumption 3 we assume that the sink node does not fail.

The network condition is classified into one of two states, either operating or failed, and is determined by the proportion of targets that are covered denoted by $C(\mathcal{G})$. For a given α-coverage requirement, $0 < \alpha \leq 1$, the *α-failure time* is the time at which $C(\mathcal{G})$ drops below α and the network transitions to a failed state. Depending on the size of the region covered and application of the WSN, we may not require 100% coverage of targets to construct a picture of the overall status. Environmental applications such as office building climate control may allow for a smaller α-coverage requirement, while applications in target tracking or surveillance may require a larger coverage requirement [2]. It may also be that 100% coverage is too costly or impractical to maintain over the life of the network. With an α-coverage requirement specified, the network's reliability $r(t;\alpha,n_1) = \Pr[C(\mathcal{G}(t)) \geq \alpha]$ is defined to be the probability that the network's coverage is at least α at time t. In the following sections we are interested in comparing the reliability of networks of varying size, resulting in n_1 appearing in the expression $r(t;\alpha,n_1)$ to denote its dependence on the number of sensors.

Assumption 4 *The initial WSN $\mathcal{G}'(0)$ is a random geometric graph (RGG) with uniform density over a bounded region $\mathcal{R} \in \mathbb{R}^2$.*

The arrangement of sensors in a WSN is typically classified as either deterministic or random [3]. With the complexity already present in (i) estimating network reliability and (ii) evaluating time-based deployment policies, Assumption 4 models the the initial network as a random geometric graph with senor nodes randomly distributed over a bounded region $\mathcal{R} \in \mathbb{R}^2$. Modeling the sensors as uniformly distributed imposes no additional loss of generality, as the results that follow hold for any density function. This removes the difficulty of designing a WSN in addition to considering these aspects, and also reflect a scenario in which a network has to be rapidly deployed in a remote area.

2.2.1 Homogeneous Network Reliability

In the absence of additional sensors being deployed in the network, the collection of sensors is homogeneous in the sense that all surviving sensors were installed at the same time and therefore have i.i.d. residual life distributions. Given the difficulty already present in estimating network reliability in this case, we first turn attention to the homogeneous network reliability using a Monte Carlo approach. Monte Carlo methods for WSN reliability evaluation give rise to a network destruction problem, an optimization problem that determines the instant of network failure given fixed sensor failure times [29]. For a network with node failures, the *destruction spectrum* is a probability distribution on the number of failed nodes required to cause network failure. The D-spectrum for a network can be estimated with steps similar to the crude Monte Carlo where the number of failed nodes corresponding to network failure is recorded instead of the time at which this occurs.

Let $s_{\alpha,i}^{n_1}$ denote the probability that in the network $\mathcal{G}$ with n_1 sensors, the i^{th} sensor failure results in $C(\mathcal{G})$ falling below α. Network reliability is then given by

$$r(t;\alpha,n_1) = \sum_{i=1}^{n_1} s_{\alpha,i}^{n_1} B(i-1;n_1,F(t)), \tag{1}$$

where $B(i-1;n_1,F(t))$ is the cumulative binomial probability of no more than $i-1$ successes in n_1 trials with probability of success $F(t)$ [34]. Although algorithms exist for computing $s_{\alpha,i}^{n_1}$ exactly, we use a Monte Carlo approach to estimate the D-spectrum which is common especially for large, complex networks [29, 35]. Therefore we use the notation $\hat{s}_{\alpha,i}^{n_1}$ to refer to the estimate of $s_{\alpha,i}^{n_1}$, leading to the reliability estimate $\hat{r}(t;\alpha,n_1)$ of $r(t;\alpha,n_1)$.

An estimate on the variance of network reliability may also be of interest, particularly if we wish to compare destruction algorithms, and is given by

$$\begin{aligned} Var(r(t;\alpha,n_1)) =& \frac{1}{M}\Big[\sum_{i=1}^{n_1} s_{\alpha,i}^{n_1}(1-s_{\alpha,i}^{n_1})B(i-1;n_1,F(t))^2 \\ &-2\sum_{j=1}^{n_1-1}\sum_{k=j+1}^{n_1} s_{\alpha,j}^{n_1}s_{\alpha,k}^{n_1}B(j-1;n_1,F(t))B(k-1;n_1,F(t))\Big], \end{aligned} \tag{2}$$

where M is the total number of replications [29]. Since we again use the D-spectrum estimate $\hat{s}_{\alpha,i}^{n_1}$, the estimate of the variance is denoted $\widehat{Var}(r(t;\alpha,n_1))$.

As mentioned previously an appealing aspect of the D-spectrum is that it is a property of the network structure and failure definition of the network, and does not depend on the lifetimes of the individual components [30]. With Assumption 1, the implication is that each of the $n_1!$ permutations of sensor failures are equally likely. Therefore instead of generating a random failure time from the distribution of T for each sensor, we can proceed to generate a random order of sensor failures. This is a key result that we revisit later towards identifying an efficient destruction algorithm on estimating the signature of the network.

2.2.2 A Generic Algorithm for Estimating the WSN's Destruction Spectrum

The first step towards estimating network reliability is now calculating the D-spectrum. Estimating the D-spectrum is outlined in Algorithm 1, based on the work in [29]. The driving component of Algorithm 1 is Step 5 of determining which sensor failure results in network failure.

In Step 5, the network is subject to a destruction process where sensors are iteratively removed from the functioning set of nodes based on the order π of sensor failures. After the failure of each sensor, the network coverage $C(\mathcal{G}(t))$ is computed and compared to the

Algorithm 1 Monte Carlo algorithm for estimating destruction spectrum

1: **function** SIGNATUREMC
2: **Set** $m_i \leftarrow 0,\ \forall\ i \in \mathcal{N}_1$.
$\triangleright$ $m_i = \#$ network failures caused by ith sensor failure
3: **Generate** $\mathcal{G}$ by simulating $(x_i, y_i) \in \mathcal{R},\ \forall\ i \in \mathcal{N}_1$.
$\triangleright$ $(x_i, y_i) =$ coordinates for sensor i
4: **Simulate** random permutation π of the sensors $\{1, 2, \ldots, n_1\}$; $\pi = (i_1, i_2, \ldots, i_{n_1})$.
5: **Find** the smallest value $i^\star \in \{1, \ldots, n_1\}$ such that $C(\mathcal{G} \setminus \{\pi(1), \ldots, \pi(i^\star)\}) < \alpha$.
$\triangleright$ The $i^\star$-th sensor failure causes network failure
6: **Set** $m_{i^\star} \leftarrow m_{i^\star} + 1$.
7: **Repeat** Steps 3–6 M times.
8: **Set** $\hat{s}^{n_1}_{\alpha,i} \leftarrow m_i/M,\ \forall\ i \in \mathcal{N}_1$.
9: **end function**

α-coverage requirement. Consider a straightforward destruction algorithm for this step. For each of the networks $\mathcal{G}\setminus\{\pi(1), \ldots, \pi(i)\}, i = 0, 1, \ldots, n_1$, implement a breadth-first search algorithm in order to identify how many of the target nodes $\mathcal{N}_2$ are reachable from the sink. Dividing this count by $|\mathcal{N}_2|$ we can determine $C(\mathcal{G})\setminus\{\pi(1), \ldots, \pi(i)\}$ based on the number of targets reached. Overall this requires $O(|\mathcal{A}|n_1)$ effort per iteration of Step 5. The destruction algorithm is implemented in each of the M replications of the Monte Carlo, which motivates the search for an efficient algorithm of finding this sensor failure of interest. A binary search method can improve the complexity of this step to $O(|\mathcal{A}|\ \log(n_1))$. In Section 2.3 we present a destruction algorithm that improves on this complexity by exploiting aspects of the D-spectrum and the network construction.

Each iteration of Step 5 returns an observation on the number of failed nodes resulting in network failure. By recording this value for every iteration we are able to obtain our estimate $\hat{s}^{n_1}_{\alpha,i}$ of the D-spectrum upon completion, and finally estimate network reliability by substituting into (1).

2.2.3 Time-Based Deployment Policies

To prolong the functioning status of the WSN we are interested in periodically deploying new sensors in the region in an effort to increase the number of functioning senors, thereby increasing network coverage. We focus on a time-based deployment policy in which n_1 sensors are initially deployed over the region $\mathcal{R}$. Every δ time units thereafter, all failed sensors in the network are replaced by deploying new sensors over $\mathcal{R}$ such that the total number of functioning sensors in the network is increased back to n_1. Such a policy is identified as an (n_1, δ) policy. By replenishing the number of functioning sensors to a constant value, the present (n_1, δ) policy assumes that we have knowledge about the number of failed sensors in the network prior to any deployment action. Similar to the initial layout of sensors, by assuming that new sensors are always deployed uniformly and independently over $\mathcal{R}$ and that each sensor's location is independent of its time to failure we arrive at the following properties.

Property 1 *For all $t \geq 0$, the WSN $\mathcal{G}'(t)$ is a RGG with uniform density over $\mathcal{R}$.*

Property 2 *For all $t \in \{k\delta : k \in \mathbb{Z}_{\geq 0}\}, |\mathcal{N}_1(t)| = n_1$.*

2.2.4 Heterogeneous Network Reliability

Towards identifying an optimal time-based deployment policy, we now analyze network reliability in the presence of a given (n_1, δ) policy. Compared to Section 2.2.1, the collection of sensors is now heterogeneous in the sense that the surviving population of sensors have different ages, and thus different residual life distributions. In [30] it was shown that the D-spectrum representation of a network remains valid in stochastic mixtures of components, provided that the components are exchangeable. With this in hand, the D-spectrum approach in the previous section can be extended to the present (n_1, δ) policy under consideration to estimate network reliability.

We refer to the time interval $[(k-1)\delta, k\delta]$ as the kth epoch, $k \in \mathbb{Z}_{\geq 0}$. Immediately after new sensors are deployed, the age X of each sensor is a random variable in the range $\{k\delta : k \in \mathbb{Z}_{\geq 0}\}$. In the presence of the (n_1, δ) policy, we are interested in the network's reliability for the infinite-horizon setting, where at the beginning of an epoch the probability distribution on the age X of a sensor does not change from one epoch to the next (i.e., there is a stable mix of sensors).

In the infinite-horizon setting, each sensor's age at the beginning of epochs $k' \in \mathbb{Z}_{>0}$ can be viewed independently as an irreducible Markov chain on the countably infinite state space $k \in \mathbb{Z}_{\geq 0}$, where state k corresponds to the sensor having age $k\delta$. In this Markov chain, each state k transitions into state $k+1$ with probability $\bar{F}((k+1)\delta)/\bar{F}(k\delta)$ and back to state 0 otherwise. This Markov chain has the unique stationary distribution

$$\rho_k = \frac{\bar{F}(k\delta)}{\sum_{j=0}^{\infty} \bar{F}(j\delta)}, \ \ k \in \mathbb{Z}_{\geq 0}, \tag{3}$$

provided that the denominator converges, in which case the Markov chain is ergodic.

Now, let $T_x \geq 0$ denote the residual life of a sensor at age $x > 0$, and denote its cdf by $F_x(t) = [F(x+t) - F(x)]/\bar{F}(x)$. Ergodicity of the Markov chain described above also implies exchangeability of the sensors at stationarity: That is, immediately after the deployment of new sensors, a subset of sensors selected at random have i.i.d. age described by the probability distribution $\Pr\{X = k\delta\} = \rho_k$, $k \in \mathbb{Z}_{\geq 0}$. The residual lifetime of such a sensor (considering the randomness in its age), is then described (see, e.g., [36]) by the cdf

$$G(t;\delta) = \mathbb{E}[F_X(t)], \tag{4a}$$

$$= \sum_{k=0}^{\infty} \frac{F(k\delta + t) - F(k\delta)}{\bar{F}(k\delta)} \rho_k, \tag{4b}$$

$$= \frac{\sum_{k=0}^{\infty} [F(k\delta + t) - F(k\delta)]}{\sum_{j=0}^{\infty} \bar{F}(j\delta)}. \tag{4c}$$

We will refer to T_X as the *stable residual life distribution* of a sensor under the (n_1, δ) policy.

Considering the above, at stationarity, the remaining life of sensors selected at random are independent and identical random variables with cdf given by (4c). Further, by Property 2, at the beginning of every epoch $k \in \mathbb{Z}_{\geq 0}$ the network contains n_1 functioning sensors and the D-spectrum of the network remains applicable. Therefore, applying (1) to the i.i.d. residual life distribution, the stable network's reliability (i.e., the probability that its coverage remains at least α after $t \geq 0$ additional time units) is given by

$$r^{\infty}(t;\alpha, n_1, \delta) = \sum_{i=1}^{n_1} s_{\alpha,i}^{n_1} B(i-1; n_1, G(t;\delta)), \tag{5}$$

where the ∞-superscript has been appended to r to denote that it applies to the infinite-horizon setting. The heterogeneous network reliability estimate is represented by $\hat{r}^{\infty}(t;\alpha, n_1, \delta)$, as it again depends on the D-spectrum estimate $\hat{s}_{\alpha,i}^{n_1}$. The variance of the (n_1, δ) policy can be estimated applying (2), again substituting the residual life ccdf $G(t;\delta)$ for $F(t)$. While we are primarily interested in the stable network reliability immediately prior to the deployment of additional sensors (i.e., at time $t = \delta$), (5) can be applied to evaluate the stable network reliability for any time $t \leq \delta$. This property maintains the ability to evaluate reliability over time, enabling system performance availability measures to be estimated as well.

Notice also that the D-spectrum is independent of the deployment interval δ. Thus, in the process of exploring the space of (n_1, δ) policies, we need only to estimate the signature once for any value of n_1 considered.

2.3 Destruction Algorithms

With a basis to estimate both the homogeneous and heterogeneous network reliability relying on the D-spectrum, we now revisit Step 5 of Algorithm 1 and search for an efficient destruction algorithm. Over the course of exploring (n_1, δ) policies it may be necessary to estimate the D-spectrum for assorted values of n_1. While there are efficiencies that can be gained as a result of Assumption 4 that are discussed later in Section 2.3.1, an improved destruction algorithm allows for a larger number of Monte Carlo replications thereby reducing the variance of our resulting reliability estimate. Towards this effort, we present Algorithm 2 that seeks to identify which sensor failure, in a predefined sequence that specifies the time of all sensor failures, causes network coverage to drop below the α-coverage requirement.

Recall that for a target to be covered it must satisfy two criteria. First the target must be within the coverage radius of a functioning sensor, and second there must be a communication path from this sensor back to the sink node. From the construction of the network $\mathcal{G}$, this equates to a directed path from the sink node to the target node using only functioning sensor nodes as internal nodes. Such a path fails as soon as one of its internal sensor node fails, and the target becomes disconnected as soon as all such paths fail. Among all directed paths from the sink to the target, define a *critical path* as one in which the time until the first failure of an internal node in the path is maximized. Thus the failure time of a critical path to a target node equals the time at which the target is no longer covered. A critical path can be similarly defined for sensor nodes, and equates to the earliest time at which a sensor node is either failed or no longer connected to the sink. With this characterization, a critical path is defined for every node $i \in \mathcal{N}$, and the failure time of this critical path is referred to as the *critical loss time*, η_i. When necessary, the critical loss time can be further distinguished as a critical sensor loss time for a node $i \in \mathcal{N}_1$, or a critical target loss time for a node $i \in \mathcal{N}_2$.

Finding a critical path to every node $i \in \mathcal{N}$ is equivalent to the maximum capacity path problem discussed by [37]. In a network with weights defined on every node, a maximum capacity path between two nodes is a path such that the weight of the smallest node on the path is maximized. Let h_i^j denote the j^{th} directed path from the sink node to node i, and $\mathcal{H}_i = \{1, 2, \ldots, H_i\}$ index the set of all directed paths from the sink node to node $i \in \mathcal{N}$. The value of a maximum capacity path to node $i \in \mathcal{N}$ is then $\max_{j \in \mathcal{H}_i}\{\min\{q_k : k \in h_i^j\}\}$. In a directed network, such as the network $\mathcal{G}$ under consideration, a maximum capacity path from a source node to every other node can be found using a slight modification to Dijkstra's algorithm while updating node labels [37]. When solving for the maximum capacity path, the label of a node is initialized as $\eta_i = 0$ for all $i \in \mathcal{N} \backslash \{0\}$, and $\eta_0 = \infty$. Nodes then have their label updated according to

$$\eta_j = \max\{\eta_j, \min\{\eta_i, q_j\}\}. \tag{6}$$

Under Assumption 3 the sink node does not fail, which is equivalent to representing the sink node as the last node to fail by $q_0 = n_1$, while target nodes are regarded in a similar fashion with $q_i = n_1$ for all $i \in \mathcal{N}_2$.

Originally introduced as a variation of the shortest path problem, the maximum capacity path is commonly defined for weights associated with every edge [37]. The network $\mathcal{G}$ can be transformed to adopt this convention by defining edge weights according to the minimum of the two adjacent nodes, with, $w_{ij} = \min\{q_i, q_j\}$ for all $(i, j) \in \mathcal{A}$. The updating of node labels in (6) can also be updated accordingly to compare the weight of the edge by $\eta_j = \max\{\eta_j, \min\{\eta_i, w_{ij}\}\}$.

A naive implementation of Dijkstra's can be accomplished in $O(|\mathcal{N}|^2)$ time, and improved to $O\big(|\mathcal{A}| + |\mathcal{N}| \log(|\mathcal{N}|)\big)$ with a heap data structure [38]. Further, the critical target loss times will be marked permanent in a non-increasing manner within Dijkstra's algorithm which means they can be sorted over the course of the algorithm, simplifying the search for the α-failure time upon completion. While possible to sort the critical loss time for all nodes, the order of critical target loss times are of particular interest as these values correspond to a change in network coverage. Therefore, let $\tilde{\eta}_{(i)}$ represent the i^{th} smallest critical target loss time for a node $i \in \mathcal{N}_2$, resulting in $\tilde{\eta}_{(1)} \leq \tilde{\eta}_{(2)} \leq \cdots \leq \tilde{\eta}_{(n_2)}$.

Using the D-spectrum to estimate network reliability we can improve the complexity further. Because $q_i \in \{1, 2, \ldots, n_1\}$ for all $i \in \mathcal{N}_1$, the labels $\eta_i, i \in \mathcal{N}$, are always in the range $\{0, 1, \ldots, n_1\}$. This feature motivates the use of Dial's implementation of Dijkstra's algorithm. In Dial's implementation the node to mark permanent in Step 5 of Algorithm 2 during an iteration can be found more efficiently by storing the temporary label of nodes in a sorted bucket structure. Initially, buckets $\{0, 1, \ldots, n_1\}$ are created with all nodes in bucket zero, except for the sink node which is placed in bucket n_1. Starting with bucket n_1, select the sink node to mark permanent, and update the label of adjacent nodes (i.e., by

Algorithm 2 Coverage Destruction

1: **function** SignatureSubroutine
2: **Initialize** $\eta_0 = \infty,\ \eta_i = 0\ \forall\ i \in \mathcal{N}\backslash\{0\}$.
3: **Initialize** $S = \emptyset, \bar{S} = \mathcal{N}$.
4: **While** $|S| < |\mathcal{N}|$.
5: **Select** node $i \in \bar{S}$ such that $\eta_i = \max\{\eta_j : j \in \bar{S}\}$.
6: **Update** $S = S \cup \{i\}$, $\bar{S} = \bar{S}\backslash\{i\}$.
7: **For** each $j : (i, j) \in \mathcal{A}$.
8: **Update** $\eta_j = \max\{\eta_j, \min\{\eta_i, q_j\}\}$.
9: **End For**.
10: **End While**.
11: **Let** $\tilde{\eta}_{(1)} \leq \tilde{\eta}_{(2)} \leq \cdots \leq \tilde{\eta}_{(n_2)}$ denote the sorted η_i-values for $i \in \mathcal{N}_2$.
12: **Find** smallest integer κ such that $\frac{n_2 - \kappa}{n_2} < \alpha$.
13: **Set** $i^* = \tilde{\eta}_{(\kappa)}$.
14: **end function**

moving to the bucket numbered with the new label value) according to (6). Continuing in this manner, the node to mark permanent at each iteration can be found efficiently as the node in the largest valued non-empty bucket. Using Dial's algorithm with ordered sensor failures, the step of finding the critical path from the sink node to every other node in the network can now be accomplished in $O(|\mathcal{N}_1| + |\mathcal{A}|)$ time [38].

The overall complexity of Algorithm 2 is also $O\big(|\mathcal{N}_1| + |\mathcal{A}|\big)$, driven by Dial's implementation of Dijkstra's in Steps 4−10. Step 2 and 3 each require $O(|\mathcal{N}|) = O(|\mathcal{N}_1| + n_2)$ time, and as a result of using modified Dijkstra's algorithm nodes are marked permanent in a non-increasing manner. Therefore the sort in Step 11 can be accomplished by simply recording the order in which target nodes are marked permanent in Step 6, and the sorting of critical target loss times does not add to the complexity. Step 12 requires $O(n_2)$ time, but can be improved to $O(1)$ time. With α known, network failure occurs when $\kappa^* = \lceil n_2(1-\alpha) \rceil$ targets are no longer covered and we can simply return $\tilde{\eta}_{(\kappa^*)}$. In any case, under the assumption that each target is initially within range of at least one functioning sensor, then $|\mathcal{A}_2| \geq n_2$, and since $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$ Step 12 (and Step 2−3) does not increase the complexity.

2.3.1 Extensions of Destruction Algorithms

In the exploration of various (n_1, δ) policies, further efficiency can be gained as a result of Assumption 4. With sensors independently and randomly located in the network we have the signature relation $s_{\alpha,i}^{n_1} = s_{\alpha,i-1}^{n_1-1}$ for $i = 1, 2, \ldots, n_1$. That is, failure of one node in a RGG with n_1 nodes yields a RGG with $n_1 - 1$ nodes. This result is also previously stated as Property 1. Therefore it is not necessary to recompute the network signature for every value of n_1 desired. Utilizing this feature allows the space of (n_1, δ) policies to be explored in a more efficient manner.

We may also be interested in the impact that α has on network lifetime as this is ultimately the criteria used to classify the network as operational. It seems reasonable to expect that for a smaller α network lifetime would be longer, and at any time t the network reliability would be higher. We can make a slight modification to Algorithm 2 to explore how large of an impact this will have. Note that the critical target loss times are independent of α. If we are interested in network reliability for various coverage requirements, one option is to first specify these various levels upfront. Then in Step 12 and 13 instead of returning a single value $\tilde{\eta}_{(\kappa^*)}$, we can easily find the α-failure time for each of these levels and update the D-spectrum estimate $\hat{s}_{\alpha,i}^{n_1}$ for each different requirement. Alternatively, we can store the entire sequence of critical target loss times, specify α upon completion and then search for the α-failure times as required. This second approach may be of more interest, particularly with respect to the D-spectrum where we are more concerned with sensor failures that result in a change in coverage. By storing the entire sequence of critical target loss times we can easily determine how sensitive the network is to the next sensor failure. In either case, the complexity of Algorithm 2 does not increase as a result of estimating network reliability for multiple coverage requirements.

2.3.2 Spanning Tree Destruction Algorithm

Thus far we have modeled the WSN as a directed network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ that includes every node (sensor and target) in the network. Since the network $\mathcal{G}'$ is smaller than $\mathcal{G}$, we might find it appealing to first work with the smaller network before expanding to the larger directed network as required. Whether we work with the network $\mathcal{G}$ or $\mathcal{G}'$, the critical path for a sensor will not change. In undirected networks such as $\mathcal{G}'$, it is known that a maximum weight spanning tree contains a maximum capacity path between all pairs of nodes in the network [39]. This property gives rise to an efficient approach for solving the destruction problem in the case of K-terminal connectivity [26, 27, 35]. Such an approach can also be adapted to our problem. If we work with the smaller undirected network $\mathcal{G}'$ and proceed with the spanning tree approach, edge weights must be defined according to $w_e = \min\{q_i, q_j\}$ for every edge $e = \{i, j\} \in \mathcal{E}$. Using Prim's Algorithm, a maximum weight spanning tree over $\mathcal{G}'$ can now be found requiring $O\big(|\mathcal{E}| + |\mathcal{N}_1| \log(|\mathcal{N}_1|)\big)$ effort [38]. Once the spanning tree is constructed we are able to find the critical loss time η_i for each sensor $i \in \mathcal{N}_1$. With the critical sensor loss times, we can consider the sensor-to-target arcs $\mathcal{A}_2$ to determine the critical loss times for all targets $i \in \mathcal{N}_2$. For each target computing η_j requires $O(|\{i \in \mathcal{N}_1 : (i, j) \in \mathcal{A}_2\}|)$; therefore the total effort required for this step is $O(\sum_{j \in \mathcal{N}_2} |\{i \in \mathcal{N}_1 : (i, j) \in \mathcal{A}_2\}|) = O(|\mathcal{A}_2|)$. From $\eta_j \in \{0, \ldots, n_1\}$ for all $j \in \mathcal{N}_2$, a bucket sort algorithm can be used to sort the critical target loss times resulting in $O(|\mathcal{N}_1| + n_2)$

effort [40].

The spanning tree approach to the D-spectrum thus requires $O\big(|\mathcal{E}|+|\mathcal{N}_1|\ \log(|\mathcal{N}_1|)+|\mathcal{A}_2|\big)$ effort. From this we can see that it is actually advantageous to work on the entire directed network $\mathcal{G}$, as it allows us to implement a more efficient algorithm to determine the critical target loss time, and in turn the D-spectrum of the network.

2.4 Optimal (n_1, δ) Policies

Section 2.2.4 provides a methodology for estimating network reliability for a given (n_1, δ) policy. We now focus on identifying values of n_1 and δ that will effectively balance cost and reliability. We assume, in the vein of *economic dependence* models in the multicomponent maintenance literature [41], that a fixed cost of $c_F > 0$ is incurred for each time at which one or more new sensors are added to the network and a variable cost of $c_V > 0$ is incurred for each new sensor added. (The fixed cost would likely be large relative to the variable cost, for instance, in a WSN that monitors a harsh/remote environment such as a glacier or another planet's atmosphere.) In the infinite-horizon setting the average cost per unit time, or long-run average cost rate, associated with an (n_1, δ) policy is given by

$$v(n_1, \delta) = \frac{c_F\{1 - [\bar{G}(\delta;\delta)]^{n_1}\} + n_1 c_V G(\delta;\delta)}{\delta}, \tag{7}$$

where the first term in the numerator is the expected fixed cost incurred (based on the probability that at least one sensor fails), the second term is the expected variable cost incurred (based on the expected number of sensor failures), and the denominator is the time between deployment actions.

We incorporate reliability into the optimization via maximizing with respect to $\omega(n_1, \delta) = r^{\infty}(\delta; \alpha, n_1, \delta)$. The resulting bi-objective optimization model is

$$\max_{n_1,\delta}\{-\upsilon(n_1,\delta), \omega(n_1,\delta)\}. \tag{8}$$

The two-variable model (8) can be approximately solved by enumerating combinations of n_1 and δ, allowing for an efficient frontier to be generated over the range of policies evaluated.

2.5 Numerical Results

We now illustrate the methodology described in Sections 2.2–2.4 in an example scenario, where the lifetime T of each sensor is distributed according to a Weibull distribution with a shape parameter $\beta = 1.5$ and scale parameter $\lambda = 10$. Sensor capabilities are defined according to a communication radius of $d_1 = 0.075$ and a sensing radius of $d_2 = 0.075$. The coverage area consists of $|\mathcal{N}_2| = 441$ targets uniformly spaced as a 21×21 grid in the region $\mathcal{R} = [0, 1] \times [0, 1]$.

Before proceeding to the reliability results there are two components of Algorithm 1 that are also worth discussing, those being the simulation of a RGG in Step 3 and the random permutation of failures in Step 4. Implementing these steps in a naive manner can result in significant effort. In a straightforward approach, a RGG of n_1 nodes can be generated in $O(n_1^2)$ time by randomly placing each node in $\mathcal{R}$, and then comparing the distance between all $\binom{n_1}{2}$ pairs of nodes to determine if an arc between the nodes is present. Given the complexity of Algorithm 2 is $O(|\mathcal{N}_1| + |\mathcal{A}|)$, the potential $O(n_1^2)$ cost is significant as we can now expect to spend more time generating a RGG than implementing a destruction algorithm. Efficient methods to generate a random graph have thus attracted a large amount of attention. In an attempt to reduce this source of complexity, Step 3 is implemented based on the technique described in [42] which generates a RGG by assigning each node in $\mathbb{R}^2$ to a bin, and then comparing nodes i and j from the appropriate bins to determine if the arc (i, j) is present. The complexity now depends on the number of bins as well, but if done appropriately the expected complexity is $O(|\mathcal{N}_1| + |\mathcal{E}|)$ [42].

The next step of simulating a random permutation of sensor failures is also worth examining further. A naive approach is to generate a failure time for each sensor from the distribution of T, then sort these values to determine the failure order. A number of available algorithms (e.g., [40]) can accomplish this in $O(n_1 \log(n_1))$ time. Instead, we use a modern version of the Fisher-Yates shuffle algorithm that generates a random permutation directly in $O(n_1)$ time [43].

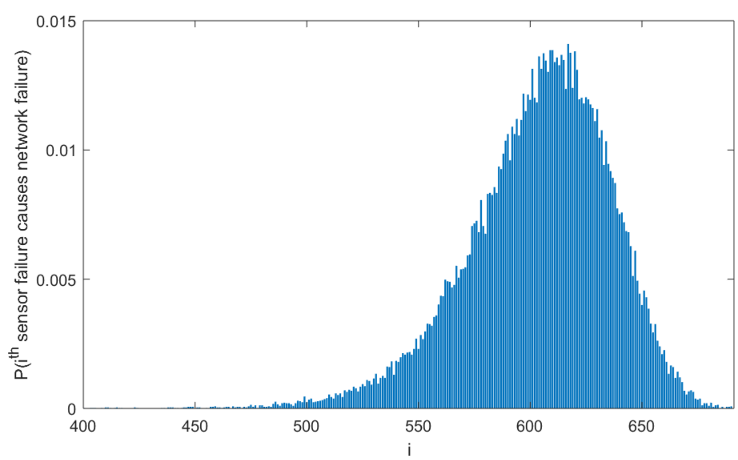

Figure 2: Plot of $\hat{s}^{900}_{0.8,i}$

The D-spectrum was estimated using Algorithm 1 with $M = 50,000$ replications for a coverage requirement of $\alpha = 0.8$ in a network consisting of $n_1 = 900$ sensors. A plot of the resulting D-spectrum estimate is illustrated in Figure 2. By the discussion at the beginning of Section 2.3.1, we can also obtain an estimate of the signature for any network containing $n_1 < 900$ sensors with no additional replications. This becomes particularly useful when evaluating time-based deployment policies, as we can now examine any (n_1, δ) policy (such that $n_1 < 900$) without re-implementing Algorithm 1. As an example, Figure 3(a) depicts the estimated D-spectrum for a network with $n_1 = 500$ sensor nodes, based on the D-spectrum estimate from the 900 node estimate from Figure 2. To illustrate the accuracy of the signature relation, we have also utilized Algorithm 1 to estimate the signature on a network with 500 sensor nodes, which is plotted in Figure 3(b).

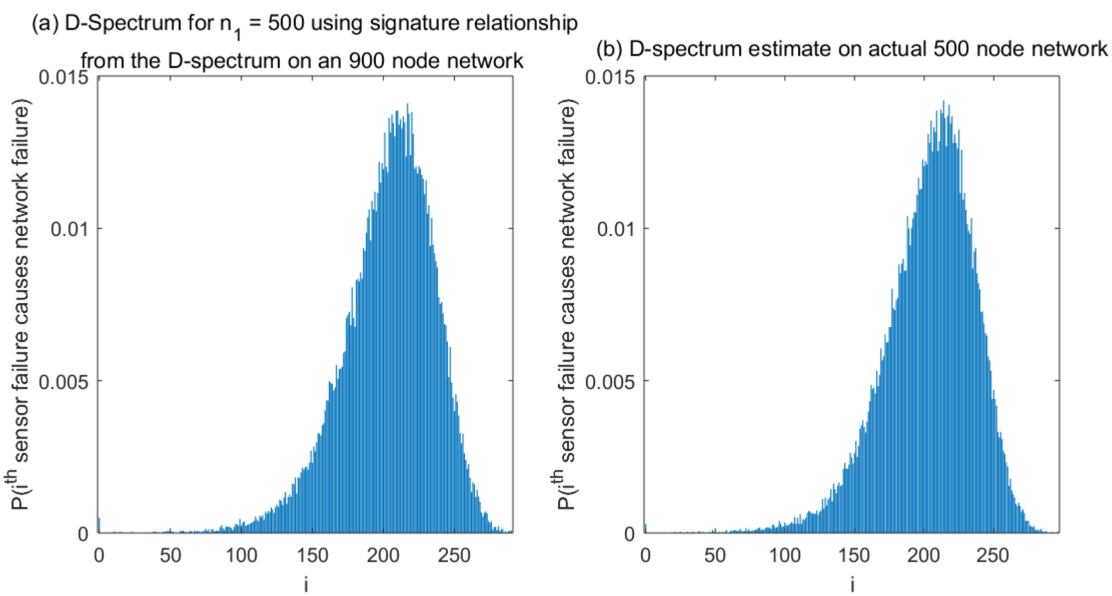

Figure 3: Comparison of D-spectrum estimates

We can now turn to reliability, and apply (1) to estimate the homogeneous network reliability if desired. Since this information is not specifically of interest in the context of comparing (n_1, δ) policies, the plot of $\hat{r}(t; \alpha, n_1)$ has been omitted. Instead, we proceed to estimate the stable network's reliability under the presence of various (n_1, δ) policies as given by (5), and the cost of the policy as given by (7). In doing so we assume a fixed cost of

$c_F = 100$ and a variable cost of $c_V = 1$. This information is plotted in Figure 4 for networks of four different selected sizes ($n_1 \in \{450, 550, 650, 750\}$), and δ evaluated over the range $(1, 10)$ at 0.1 unit intervals. Additionally, δ at specific intervals has been identified on the plot. If there are factors that impose limitations on the network size (e.g., n_1 must be a multiple of 10) or the deployment interval, then Figure 4 can be particularly valuable in comparing the performance of various policies. For example, both the $(550, 4.5)$ policy and the $(650, 5.9)$ policy yield a stable network reliability of 0.85. To meet a stable network reliability requirement above 0.85 the deployment interval for each policy must decrease, at which point the $(650, \delta < 5.9)$ policy dominates any $(550, \delta < 4.5)$ policy.

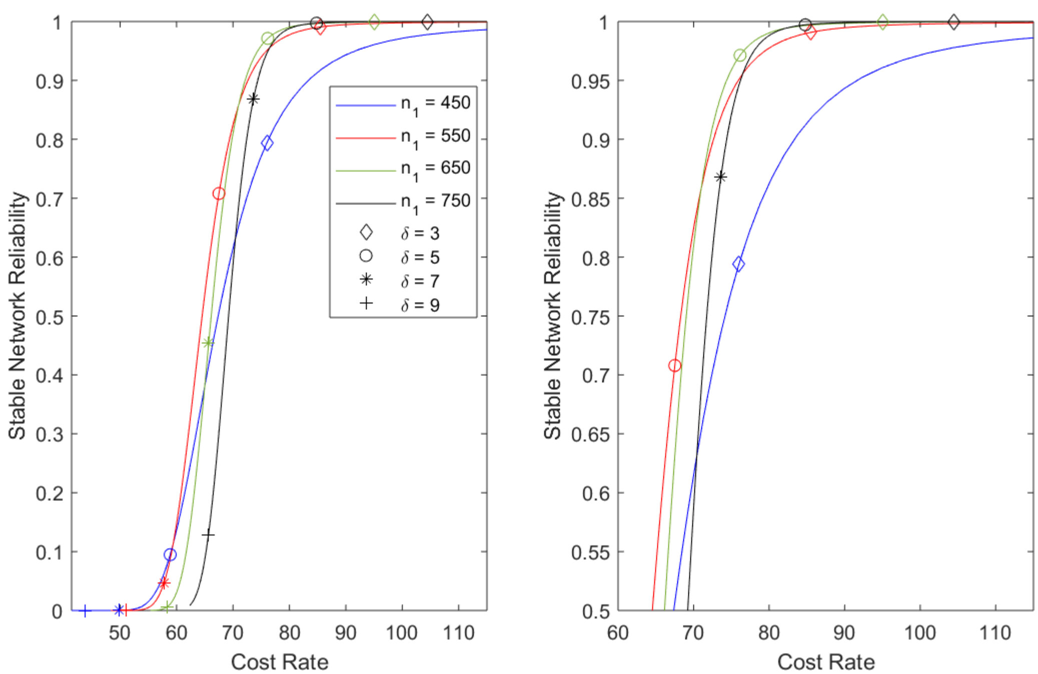

Figure 4: Plot of stable network reliability

In the current context of a RGG and sensors randomly deployed, we can assume to freely select both n_1 and δ over a continuous range. Therefore, Figure 5 is of more significance as it illustrates an efficient set of policies over a continuous range of (n_1, δ) of interest. The fairly intuitive result behind Figure 5 is that to satisfy a larger requirement on the stable network reliability, in general the network should contain a larger number of sensors and the deployment interval δ should be smaller. We can also observe that near 100% reliability can be achieved with $n_1 \approx 675$ with a deployment interval of $\delta \approx 4.5$. As we deviate from this policy by either adding more sensors or decreasing the deployment interval, the cost rate increases significantly.

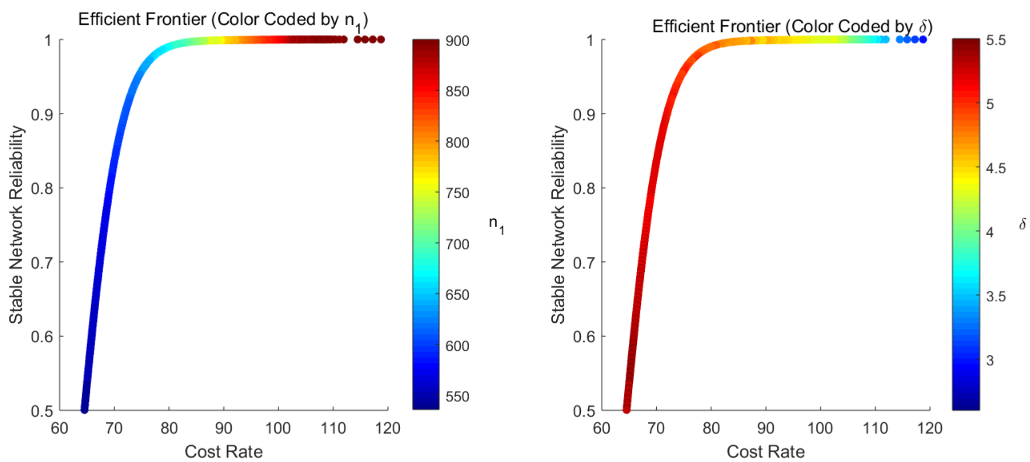

Figure 5: Efficient Frontier for $\alpha = 0.8$

Using the methods previously described to generate a RGG and new failure order each iteration, along with the destruction algorithm from Section 2.3, estimating the D-spectrum for a 900 node network required approximately 374 seconds (accomplished with c++ on an Intel(R) Core i7-6600U CPU with a 2.60 GHz processor and 16 GB of RAM). This D-spectrum estimate is then used to evaluate the set of (n_1, δ) policies for $n_1 \in \{500, 501, \ldots, 900\}$ and $\delta \in \{1.0, 1.1, \ldots, 10.0\}$ and plot the efficient frontier in Figure 5. This process is far more computationally expensive, requiring approximately $5,061$ seconds (~ 84 minutes).

Delineating the time between these two step allows us to clearly see the benefit of modeling assumption 4, and the advantage of using the D-spectrum relation $s_{\alpha,i}^{n_1} = s_{\alpha,i-1}^{n_1-1}$

to estimate the signature of smaller networks. While we could use Algorithm 1 to estimate the D-spectrum for each network size under consideration, doing so would add up to $374 \times (900 - 500) = 149,600$ seconds (~ 41.5 hours) to the overall computation time. Even though we expect the time required to estimate the D-spectrum for smaller networks to decrease (take Figure 3(b) for example, which only required 163 seconds to estimate), the additional computation time by repetitively estimating the D-spectrum remains significant (using this estimate the additional time is approximately 18 hours).

Finally, we may be interested in how sensitive the efficient frontier is to various parameters (e.g., c_F, c_V, β, λ). Figure 6 plots the efficient frontier for two different α-coverage requirements: the original efficient frontier for $\alpha = 0.8$, along with the new efficient frontier for $\alpha = 0.9$. This plot can help illustrate the robustness of various policies and the improvement in network performance for a minor cost increase. Consider the $(600, 5)$ policy, which incurs a cost rate of 71.8. For a coverage requirement of $\alpha = 0.8$ the corresponding stable network reliability is 0.897, while for a coverage requirement of $\alpha = 0.9$ the stable network reliability drops significantly to 0.678. Clearly, the same (n_1, δ) policy will have a smaller stable network reliability for a larger coverage requirement. But now consider this relationship with respect to the $(675, 5)$ policy, which incurs a cost rate of 78.3. For a coverage requirement of $\alpha = 0.8$ the corresponding stable network reliability is now 0.984, and for a coverage requirement of $\alpha = 0.9$ the stable network reliability is 0.922. Thus by increasing the number of nodes in the network (at a minor increase to the policy cost) we can implement a policy that not only meets an α-coverage requirement of 0.8 with high probability, but also achieve a coverage requirement of 0.9 with high probability.

A similar process can be used to explore the impact of changing the associated costs of

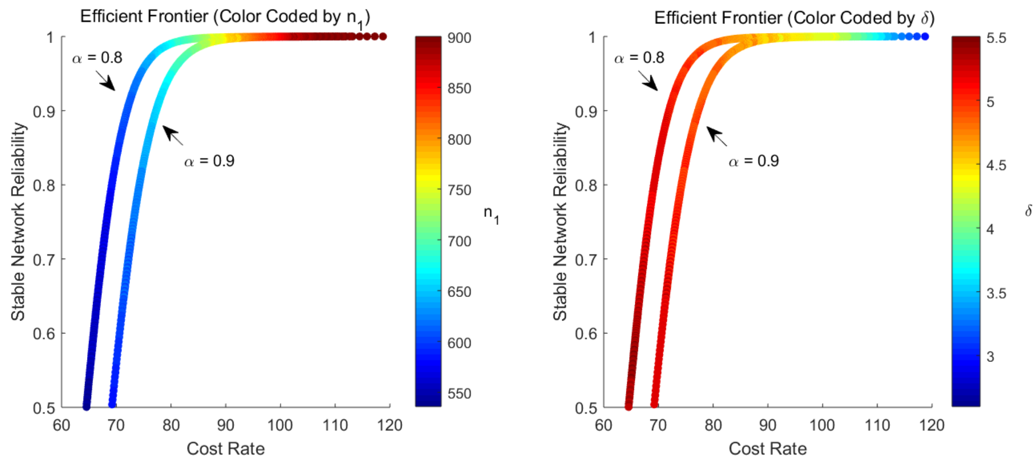

Figure 6: Efficient Frontier for different α-coverage requirements

deployment actions or the sensor failure distribution parameters. The change to the efficient frontier in each scenario is similar to that illustrated in Figure 6, the general shape of the curve remains the same but is shifted based on the direction of the parameter that is altered.

2.5.1 Confidence Interval on Stable Network Reliability

From the discussion in Section 2.2.1 we can also obtain an estimate on the network reliability variance, which in turn can be used to construct a confidence interval. Computing the confidence interval halfwidth will also help compare the performance of different destruction algorithms, illustrating the improvement that Algorithm 2 (using Dial's implementation) offers. With the variance in (2) a function of the number of replications M, the more replications we dedicate towards estimating the D-spectrum we can expect a tighter confidence interval.

To show the significance of this improvement, we consider a $n_1 = 650$ node network as a test instance. The D-spectrum is estimated using both Dial's implementation of Algorithm 2 as previously described, and also by using a naive $O(|\mathcal{N}|^2)$ implementation of Dijkstra's algorithm. For Dial's implementation we use $M = 50,000$ replications, while for Dijkstra's Algorithm we set $M = 20,000$. These values were selected so that the total time dedicated towards estimating the D-spectrum from the two methods was approximately equal.

For each D-spectrum estimate we can again compute the stable network reliability for various (n_1, δ) policies, while in addition estimating the corresponding halfwidth. The 95% confidence interval halfwidth on the stable network reliability for the $(650, \delta)$ policy is plotted in Figure 7. The improvement in the confidence interval halfwidth is most notable for $\delta \in (5, 8)$. If we revisit Figure 4, this range is also where a change in δ results in a significant change to the stable network reliability. Thus, by using Dial's implementation in Algorithm 2 we can perform over twice as many replications in the same amount of time compared to the traditional Dijkstra's algorithm, which in turn results in a confidence interval halfwidth that is twice as small compared to the original Dijkstra's estimate.

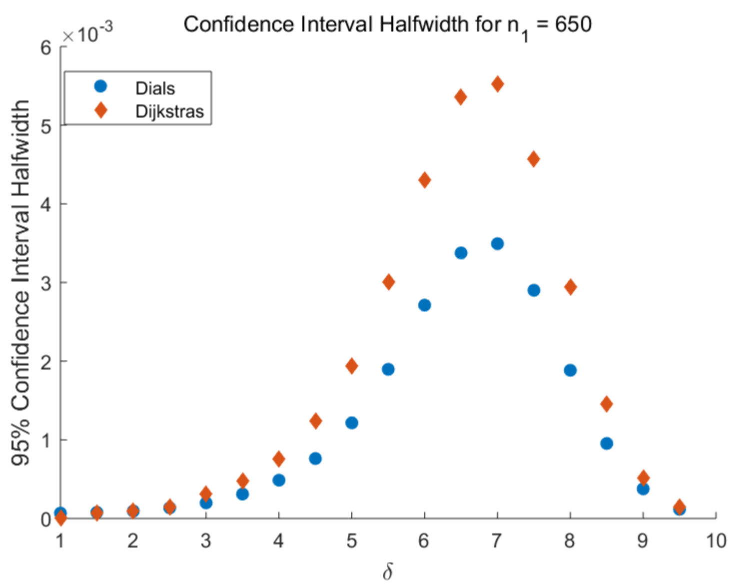

Figure 7: Confidence Interval Halfwidth Comparison

It is also interesting to note that while the total time estimating the D-spectrum between the two methods is approximately the same, there are different steps of Algorithm 1 that dominate the complexity. For this purpose we focus primarily on Step 3 and Step 5. As previously presented, generating a RGG of n_1 nodes using [42] results in an expected complexity of $O(|\mathcal{N}_1| + |\mathcal{E}|)$, while the complexity of Algorithm 2, which accomplishes Step 5, is $O(|\mathcal{N}_1| + |\mathcal{A}|)$. With $|\mathcal{A}| = 2|\mathcal{E}| + |\mathcal{A}_2|$, the complexity of these two steps is relatively balanced. However when Dijkstra's algorithm is used as the destruction algorithm, the $O(|\mathcal{N}|^2)$ now becomes a large source of complexity and significantly more time is dedicated to Step 5. Thus, by using Algorithm 2 we are performing a larger number of replications while actually spending less time in this step of the destruction spectrum algorithm.

2.5.2 Verification of Stable Network Reliability

The stable network reliability is derived from the stable residual life distribution of a sensor, as given in (4c). This is the long-run residual life distribution which is based on new sensors being deployed every δ time units. Since this applies in the infinite-horizon setting, a compelling question that arises is how long it takes to reach this steady state behavior.

To investigate this we can utilize a crude Monte Carlo simulation that implements the given (n_1, δ) policy, and check the network status at various times over the length of the simulation. This was accomplished for the $(650, 5.6)$ policy, with the resulting estimated transient network reliability illustrated in Figure 8. The stable network reliability estimated using the methodology in Section 2.2.4 is also plotted in Figure 8.

From Figure 8 we can observe that the stable network reliability is reached early on in the simulation, after the second or third deployment action. This helps demonstrate that while the stable network reliability is built on a long-run horizon, it is reached fairly early on

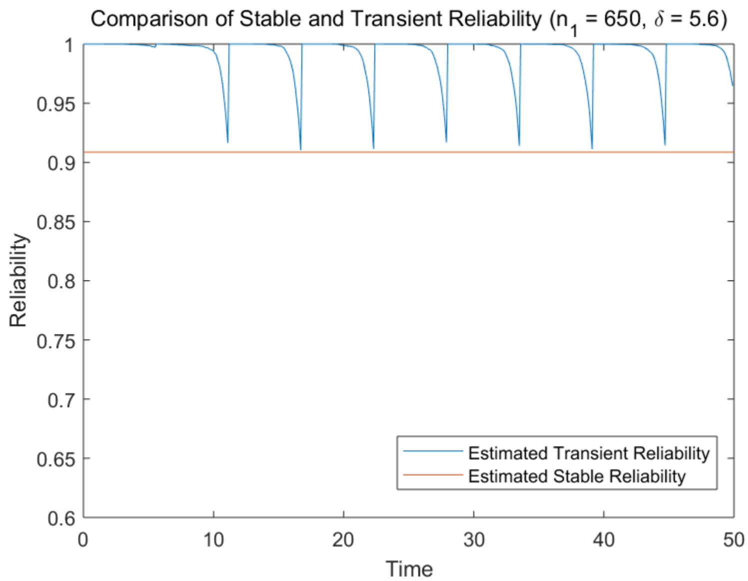

Figure 8: Verification of Stable Network Reliability

the process of the (n_1, δ) policy. While the transient Monte Carlo simulation is informative, particularly in plotting the change in reliability over time, it is far more computationally expensive and not conducive to optimizing time-based deployment policies. The transient reliability data in Figure 8 is based on 10,000 replications, and required 1.7 hours. Additionally, this simulation evaluates a single value of n_1 and δ. The transient Monte Carlo simulation therefore quickly becomes intractable, particularity if we desire to evaluate the range of polices necessary to generate an efficient frontier similar to Figure 5.

2.5.3 Verification of Long-Run Cost Rate

Similarly, the Monte Carlo simulation can also help verify the long-run cost rate estimated by (7). Based on the simulation of the $(650, 5.6)$ policy in Figure 8, the estimated long-run cost rate is 70.3. This is smaller than the estimate using the stable life distribution in (7), which results in an estimated cost rate of 72.5. The difference in these estimates is attributed to the very first deployment action in the Monte Carlo simulation. Referring back to Figure 8, this is the "cheapest" deployment event in the sense that the fewest number of sensors have failed at the time of the first deployment event compared to those that occur later, which lowers the average cost rate slightly. If we omit the cost of the first deployment action each replication (i.e., only calculate the cost once the steady state behavior has been reached), we observe an average cost rate of 72.25 from the Monte Carlo simulation.

2.5.4 Multi-State Network

In Section 2.2 the network was characterized into either an operating for failed state. There are numerous applications in which we may be interested in defining one or more intermediate states to reflect a partial degradation in network performance. An extension of the destruction spectrum to multi-state networks is discussed in [44], which can also be addressed with the current modeling framework. Since the state of the network is dependent upon a coverage requirement, multiple network states can be defined by multiple coverage levels where the state of the network is now based on network coverage falling within a given range. For example a three state network can be defined in which State 1 corresponds to $C(\mathcal{G}) \geq \alpha$, State 2 (intermediate state) in which $\alpha' \leq C(\mathcal{G}) < \alpha$, and State 3 in which $C(\mathcal{G}) < \alpha'$. For a given (n_1, δ) policy we are now interested in the probability that the network is in each of the given states. The probability the network is in State 1 can be estimated simply by $\Pr(\text{State 1}) = r^\infty(\delta; \alpha, n_1, \delta)$. Similarly, the probability the network is in State 3 can be estimated by $\Pr(\text{State 3}) = 1 - r^\infty(\delta; \alpha', n_1, \delta)$. The probability the network is in the intermediate State 2 can now be estimated by $\Pr(\text{State 2}) = 1 - \Pr(\text{State 1}) - \Pr(\text{State 3}) = r^\infty(\delta; \alpha', n_1, \delta) - r^\infty(\delta; \alpha, n_1, \delta)$. Therefore, the additional work required in a multi-state model corresponds to estimating network reliability for different coverage requirements.

An example for a three state network is illustrated in Table 1, where State 1 is defined by $C(\mathcal{G}) \geq 0.9$, State 2 by $0.8 \leq C(\mathcal{G}) < 0.9$, and State 3 by $C(\mathcal{G}) < 0.8$. While it is straightforward to calculate each state probability for the entire range of (n_1, δ) policies explored, the results for a smaller subset of policies are provided. Comparing the multi-state performance is of particular interest when a decision must be made to select a specific policy to implement. For example if we are comparing the $(676, 4.8)$ policy with the $(651, 5.1)$ policy, the first policy achieves a coverage of 0.9 with higher probability but is also more costly. We may also consider the second policy since it is less costly but still achieves a coverage of 0.9 with fairly high probability, and in the event coverage drops below the first coverage level is likely to be in the intermediate state. Notice that the data required to construct Table 1 is also previously illustrated in Figure 6, and the table presents similar information in a slightly different manner. Modeling a larger number of network states is also possible through the introduction of new coverage levels, at the expense of an additional set of reliability calculations for each new state.

Table 1: Multiple Network States

(n_1, δ) Policy	Cost Rate	Pr(State 1)	Pr(State 2)	Pr(State 3)
$(753, 4.5)$	88.78	0.995	0.004	0.001
$(676, 4.8)$	79.78	0.946	0.044	0.010
$(651, 5.1)$	75.59	0.853	0.113	0.034
$(615, 5.0)$	73.13	0.752	0.180	0.069
$(569, 5.1)$	68.54	0.452	0.316	0.231
$(555, 5.3)$	66.17	0.282	0.339	0.379

It is also of interest to investigate the impact a multi-state network has on optimizing a policy with respect to multiple coverage levels. With this motivation, we compare the similarity of the actual set of policies on each of the efficient frontiers in Figure 6. Based on the results in Figure 6, from the entire set of policies that are on the efficient frontier for at least one of the coverage requirements, 85% of these policies are on both efficient frontiers. This implies with high probability that if we select an efficient (n_1, δ) policy to achieve a coverage requirement of 0.8, this is also an efficient policy to achieve a coverage requirement of 0.9. Additionally, if we select some (n_1, δ) policy that is on the efficient frontier for a coverage requirement of 0.8 but not 0.9, the improvement (with respect to reliability) from deviating from this policy to an efficient policy for the 0.9 coverage requirement is negligible.

2.6 Conclusion

As technology advances and becomes more affordable, WSNs are able to be integrated into an increasing number of applications. While the deployment of a WSN is the initial concern, the long-run operating cost is an important factor to consider. This is true in terms of designing a network to meet requirements in addition to ensuring any maintenance policy preserves network functionality without an excessive cost. Existing research has emphasized methods to extend network lifetime, but does not focus on analyzing a maintenance policy with respect to network reliability.

Towards this goal, we have contributed an optimization model that determines optimal time-based deployment policies balancing cost and reliability. Of interest from this model is the inclusion of the destruction spectrum to evaluate policies, as the destruction spectrum is independent of the deployment interval parameter δ. This aspect helps decouple the complexity of estimating the destruction spectrum necessary to evaluate reliability, and evaluating the network's reliability in the presence of a given time-based deployment policy. Finally, we have presented a destruction algorithm to efficiently estimate the destruction spectrum, and illustrated the performance of the optimization model through a computational example.

In the network model we have focused on sensor failures that are identically distributed. One possible direction for extending this work involves the modeling of multiple sensor failure distributions. The incorporation of multiple failure distributions can be attractive, for example, when modeling the energy hole phenomena in which sensor nodes located closer to the sink node are relied on more often to route information through the network. As a result these nodes consume energy at a faster rate which leads to a shorter expected lifetime [45]. The destruction spectrum approach from Section 2.2 can be adapted to address this scenario (see [46]), with the drawback that the dimension of the signature increases.

Along the same direction, future work might consider the impact of load dependent failures. As sensor nodes fail in the network, messages must be routed along different paths to reach the sink. This inevitably results in various sensor nodes being relied upon in a larger capacity to route information, which can cause an increase in energy use and faster failure rate.

Section 2.5.4 discussed the impact of modeling multiple network states. A related version of this extension is to consider multi-state sensors. This problem variant introduces several additional sources of complexity as we are now concerned not only with the initial capability of a sensor and the total number of functioning sensors in the network, but the number of sensors in each state and the capability of a sensor in a given state. A model must also be incorporated to reflect the transition of a sensor among the various states. Investigating how these components can be incorporated into a reliability estimate, and the impact they have on the current optimization model is a compelling problem to explore.

Another direction is to consider a deterministic network topology. The assumption that sensor nodes are always randomly deployed allowed us to leverage the D-spectrum relationship between networks of different sizes, saving a large amount of computation time. However in many applications we can control the network topology more precisely by locating sensors at specific points. While many of the results in Sections 2.2–2.4 remain applicable in this case, it is not clear what/if there is a relationship between the D-spectrum for different networks, or if the D-spectrum can be found in a more efficient manner due to a deterministic topology.

We have also primarily considered a time-based deployment policy, in which new sensors are deployed every δ time units. Instead of scheduling a deployment based on time intervals, an improved policy could consider the condition of the network as well, such as the current number of failed sensors or the number of targets covered. Given the stochastic nature of the network evolution and the potentially enormous state and maintenance decision space, this modeling approach may be more amenable to approximate dynamic programming.

References

[1] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society*, vol. 3. IEEE, 2001, pp. 1380–1387.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] S. Debnath, A. K. Singh, and A. Hossain, "A comprehensive survey of coverage problem and efficient sensor deployment strategies in wireless sensor networks," *Indian Journal of Science and Technology*, vol. 9, no. 45, pp. 1–6, 2016.

[4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, 2002.

[5] D. Deif and Y. Gadallah, "A comprehensive wireless sensor network reliability metric for critical internet of things applications," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, p. 145, 2017.

[6] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems.* New York, NY: ACM, 2003, pp. 28–39.

[7] L. M. S. De Souza, H. Vogt, and M. Beigl, "A survey on fault tolerance in wireless sensor networks," *Interner Bericht. Fakultät für Informatik, Universität Karlsruhe*, 2007.

[8] N. Li and J. C. Hou, "Flss: a fault-tolerant topology control algorithm for wireless networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking.* ACM, 2004, pp. 275–286.

[9] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings 22nd International Conference on Distributed Computing Systems Workshops.* IEEE, 2002, pp. 575–578.

[10] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 1, pp. 89–124, 2005.

[11] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 1, pp. 5:1–5:39, Feb. 2009.

[12] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.

[13] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2. IEEE, 2003, pp. 1293–1303.

[14] A. A. Abbasi, M. F. Younis, and U. A. Baroudi, "Recovering from a node failure in wireless sensor-actor networks with minimal topology changes," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 256–271, 2012.

[15] P. K. Sahoo and J.-P. Sheu, "Limited mobility coverage and connectivity maintenance protocols for wireless sensor networks," *Computer Networks*, vol. 55, no. 13, pp. 2856–2872, 2011.

[16] G. Wang, G. Cao, T. La Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 4. IEEE, 2005, pp. 2302–2312.

[17] S. Parikh, V. M. Vokkarane, L. Xing, and D. Kasilingam, "Node-replacement policies to maintain threshold-coverage in wireless sensor networks," in *2007 16th International Conference on Computer Communications and Networks.* IEEE, 2007, pp. 760–765.

[18] H. M. Almasaeid and A. E. Kamal, "On the minimum k-connectivity repair in wireless sensor networks," in *2009 IEEE International Conference on Communications.* IEEE, 2009, pp. 1–5.

[19] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Networks*, vol. 14, no. 3, pp. 347–355, 2008.

[20] X. Han, X. Cao, E. L. Lloyd, and C.-C. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 5, pp. 643–656, 2009.

[21] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.

[22] J. L. Bredin, E. D. Demaine, M. Hajiaghayi, and D. Rus, "Deploying sensor networks with guaranteed capacity and fault tolerance," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2005, pp. 309–319.

[23] J. S. Provan and M. O. Ball, "The complexity of counting cuts and of computing the probability that a graph is connected," *SIAM Journal on Computing*, vol. 12, no. 4, pp. 777–788, 11 1983.

[24] M. O. Ball, C. J. Colbourn, and J. S. Provan, "Network reliability," *Handbooks in Operations Research and Management Science*, vol. 7, pp. 673–762, 1995.

[25] Q. Yang and Y. Chen, "Monte carlo methods for reliability evaluation of linear sensor systems," *IEEE Transactions on Reliability*, vol. 60, no. 1, pp. 305–314, 2011.

[26] Y. Shpungin, "Networks with unreliable nodes and edges: Monte carlo lifetime estimation," *Applied Mathematics and Computer Science*, vol. 27, pp. 168–173, 2007.

[27] T. Elperin, I. Gertsbakh, and M. Lomonosov, "Estimation of network reliability using graph evolution models," *IEEE Transactions on Reliability*, vol. 40, no. 5, pp. 572–581, Dec 1991.

[28] H. Cancela and M. El Khadiri, "A recursive variance-reduction algorithm for estimating communication-network reliability," *IEEE Transactions on Reliability*, vol. 44, no. 4, pp. 595–602, 1995.

[29] I. B. Gertsbakh and Y. Shpungin, *Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo*. Boca Raton, FL: CRC press, 2016.

[30] J. Navarro, F. J. Samaniego, N. Balakrishnan, and D. Bhattacharya, "On the application and extension of system signatures in engineering reliability," *Naval Research Logistics*, vol. 55, no. 4, pp. 313–327, 2008.

[31] H. Wang, "A survey of maintenance policies of deteriorating systems," *European Journal of Operational Research*, vol. 139, no. 3, pp. 469–489, 2002.

[32] L. Bertling, R. Allan, and R. Eriksson, "A reliability-centered asset maintenance method for assessing the impact of maintenance in power distribution systems," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 75–82, 2005.

[33] Y. Shang, W. Wu, J. Liao, G. Jianbo, J. Su, W. Liu, and Y. Huang, "Stochastic maintenance schedules of active distribution networks based on monte-carlo tree search," *IEEE Transactions on Power Systems*, 2020.

[34] F. J. Samaniego, "On closure of the ifr class under formation of coherent systems," *IEEE Transactions on Reliability*, vol. R-34, no. 1, pp. 69–72, April 1985.

[35] I. Gertsbakh and Y. Shpungin, "Combinatorial approaches to monte carlo estimation of network lifetime distribution," *Applied Stochastic Models in Business and Industry*, vol. 20, no. 1, pp. 49–57, 2004.

[36] M. Finkelstein and J. Vaupel, "On random age and remaining lifetime for populations of items," *Applied Stochastic Models in Business and Industry*, vol. 31, no. 5, pp. 681–689, 2015.

[37] M. Pollack, "Letter to the editor—the maximum capacity through a network," *Operations Research*, vol. 8, no. 5, pp. 733–736, 1960.

[38] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications.* New Jersey: Prentice Hall, 1993.

[39] T. Hu, "Letter to the editor—the maximum capacity route problem," *Operations Research*, vol. 9, no. 6, pp. 898–900, 1961.

[40] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms.* Cambridge, Massachusetts: MIT Press, 2009.

[41] R. Dekker, R. E. Wildeman, and F. A. Van der Duyn Schouten, "A review of multi-component maintenance models with economic dependence," *Mathematical Methods of Operations Research*, vol. 45, no. 3, pp. 411–435, 1997.

[42] D. Funke, S. Lamm, U. Meyer, M. Penschuck, P. Sanders, C. Schulz, D. Strash, and M. von Looz, "Communication-free massively distributed graph generation," *Journal of Parallel and Distributed Computing*, vol. 131, pp. 200–217, 2019.

[43] R. Durstenfeld, "Algorithm 235: random permutation," *Communications of the ACM*, vol. 7, no. 7, p. 420, 1964.

[44] I. B. Gertsbakh and Y. Shpungin, "Stochastic models of network survivability," *Quality Technology & Quantitative Management*, vol. 9, no. 1, pp. 45–58, 2012.

[45] J. Li and P. Mohapatra, "An analytical model for the energy hole problem in many-to-one sensor networks," in *IEEE Vehicular Technology Conference*, vol. 62, no. 4, 2005, p. 2721.

[46] F. P. Coolen and T. Coolen-Maturi, "Generalizing the signature to systems with multiple types of components," in *Complex Systems and Dependability.* Berlin: Springer, 2013, pp. 115–130.

3 Myopic Condition-Based Deployment Policies for Reliable Wireless Sensor Networks

Abstract

Wireless sensor networks (WSNs) consist of a set of sensors distributed over a region of interest that monitor and report on conditions within the network. Network reliability of region coverage is often an important performance metric, as the status of the network degrades over time due to sensor failures. To facilitate network operation over a prolonged period time, failed nodes may be replaced or new sensors deployed to re-establish network capability. We explore a condition-based sensor deployment policy, in which new sensors are periodically deployed based on an observed network state. The destruction spectrum (D-spectrum) has been utilized to estimate network reliability, and offers several advantages over a traditional Monte Carlo approach. While the D-spectrum is a function of the network structure, or the number of sensors and their distribution throughout the network, we discuss how the D-spectrum can be incorporated into a model that estimates reliability in the presence of a condition-based sensor deployment policy. This model is then demonstrated by evaluating various policies with respect to the resulting reliability for region coverage. Finally, the performance of these policies is compared to a simpler time-based sensor deployment strategy.

3.1 Introduction

Wireless Sensor Networks (WSNs) are commonly characterized by a large number of low-cost sensor nodes operating in a cooperative manner to monitor a region of interest. Additionally, WSNs require little infrastructure or supporting resources (e.g., physical wire connection) for sensors to route information through the network [1]. These features enable WSNs to be quickly established by randomly deploying sensors over a target location, which may also be necessary when operating in harsh or difficult to access terrain [2].

Over the course of network operation, sensors consume a finite power supply while monitoring the surrounding region and communicating with nearby sensors [3]. Once this power supply is consumed the sensor fails and no longer contributes to network operation. The lifetime of a sensor can be further accelerated by software or hardware complications, which may arise as a result of external (e.g., environmental) factors [1]. While the WSN can likely withstand a few sensor failures with minor impact to network capability, as a larger number of sensors fail the WSN becomes increasingly degraded.

Several different methods have been explored to prolong network lifetime in the presence of sensor failures. Topology control algorithms commonly aim to extend sensor lifetime by managing power consumption. One approach is to modify the communication radius based on the distance of nearby sensors, as a smaller communication range between sensors requires less energy [4]. There may also be redundant sensors in the network that provide little additional coverage or communication capability. In this situation a sleep/wake schedule can be used to turn sensors on and off as necessary, allowing a sensor to conserve energy until needed [5]. Another approach is through the introduction of one or more mobile sensors in the network to reposition sensors over time as necessary [6, 7]. However, there is a significant cost that accompanies this mobile capability, as the cost of such sensors can be significant compared to static sensors [3, 6]. Additionally, the WSN may be in an environment that is not conducive to sensor mobility, such as a forest or steep mountain side.

The use of topology control algorithms and mobile sensor nodes aim to extend network lifetime. That is, determining a policy to maximize the lifetime of a given WSN deployment. To enable the long-term operation of a WSN, we must eventually consider deploying new sensors in the network. The objective of deploying new sensors can be directed towards restoring a level of network coverage, and/or improving sensor communication capability. In [8] and [9] the deployment of new sensors is addressed, in addition to seeking a policy that deploys the fewest number of new sensors. This objective adds further complexity to the search for a deployment policy, as problems related to optimal node placement commonly fall in the NP-Hard class of problems [10]. It may also be difficult to locate sensors at a specific location, particularly if we are forced into a random deployment of sensors due to the operating environment.

Whereas the previous focus has primarily been on the deployment of sensors at a single point in time, in this work we consider a problem where the decision to deploy new sensors is made sequentially over a number of time periods. In doing so, we address the frequency with which new nodes are deployed in the network and the associated cost. Further, throughout the previously mentioned topology control algorithms, introduction of mobile sensor nodes, and single-stage sensor deployment, the focus has been on extending network lifetime or maintaining a coverage/connectivity requirement. We focus on evaluating a node deployment policy with respect to network reliability which commonly fall in the #P-Complete class of problems [11], and are therefore routinely estimated by approximate solution methods such as a Monte Carlo simulation.

In the following section, we discuss a condition-based node deployment model where the deployment of new sensors is based on an observed state of the WSN. The objective is to determine a sensor deployment policy that results in a highly reliable network given a fixed budget available. A Monte Carlo simulation can be used to evaluate a given condition-based deployment policy (CBDP), but improving upon and optimizing a policy is a more challenging task. We illustrate how the network destruction spectrum (D-spectrum), or signature, can be used to estimate reliability in the presence of a CBDP, alleviating some of the difficulties encountered in network reliability problems. The model is then illustrated through an example for various policies.

3.2 Problem Formulation and Methodology

Consider a WSN $\mathcal{G}$ that is comprised of a sink node and a collection of sensor nodes. These sensor nodes are deployed throughout some region of interest $\mathcal{R}$, which is partitioned into a number of smaller subregions $\{1, 2, \ldots, n_r\}$. The main tasks of a sensor node are communicating with neighboring sensors to route information through the network directed toward the sink node, in addition to monitoring nearby targets. These sensor capabilities are defined by a communication radius $d_1 > 0$, and a monitoring radius $d_2 > 0$.

Due to the failure of sensors the WSN evolves over time, impacting the ability of sensors to communicate with each other and diminishing the collection of targets covered. For a target to be covered at any given time it must be within the coverage radius of a functioning sensor, and there must exist a communication path from this monitoring sensor bask to the sink node. At time $t \geq 0$, the network $\mathcal{G}$ consists of sensors that have been deployed in the network and remain functioning at time t. The condition of the network is then defined in relation to network coverage, $C(\mathcal{G})$, and represents the proportion of targets in the network currently covered.

To maintain adequate coverage over a prolonged period of time, new sensors are deployed in the WSN. First, the network is observed and degraded portions of the network can be detected, which informs the deployment of new sensors. It may be impractical or costly to constantly monitor the state of the network [12], but it is assumed that every δ time units the network can be observed. The time intervals between observations now correspond to a series of missions, where mission m refers to the period of time between $m\delta$ and $(m+1)\delta$. If one or more sensors are deployed in the network, a fixed cost c_F is incurred in addition to a variable cost c_V per sensor deployed. It is assumed that all sensors are deployed with an independent and identically distributed (i.i.d.) life distribution, F, and that all sensor capabilities are identical.

The observed state of the network is denoted $S_m = (S_{m1}, S_{m2}, \ldots, S_{mn_r})$, where S_{mi} is the number of sensors functioning in subregion $i \in \mathcal{R}$ at the beginning of mission m. After the network is observed, a decision $x_m = (x_{m1}, x_{m2}, \ldots, x_{mn_r})$ is made on how new sensors are deployed in the network, where x_{mi} is the number of sensors deployed to subregion $i \in \mathcal{R}$ during mission m. Due to the difficulty encountered when attempting to deploy a sensor to a specific coordinate location, the initial deployment of sensors, along with all future deployments, is random within a subregion. A sensor deployment decision is faced repeatedly over a series of missions, and the decision made during mission m may impact the decision of how sensors are deployed in missions $m' > m$. However given the stochastic nature of sensor failures and the potentially enormous state and deployment decision space, we focus on a myopic condition-based deployment policy (M-CBDP) that focuses on the impact on reliability for the current mission.

3.2.1 Myopic Condition-Based Sensor Deployment

In the myopic formulation of the condition-based sensor deployment problem, a fixed budget β is available each mission and the objective is to maximize the probability of mission success. An individual mission is successful if network coverage over the duration of the mission satisfies a given coverage requirement, α. Equivalently, mission m is successful if coverage at the end of the mission (time $(m+1)\delta$) is at least α. The reliability of the network during mission m is defined as the probability the coverage requirement is satisfied for the duration of the mission, and is denoted $R(S_m, x_m)$ if we observe network state S_m and deploy sensors according to action x_m. The objective in the myopic condition-based sensor deployment problem is therefore

$$\max R(S_m, x_m), \tag{9}$$

subject to a constraint that the cost of deploying sensors, $c_F + c_V x_m$, not exceed the budget available.

Selecting an optimal action requires evaluating (9) to determine network reliability. As previously mentioned, network reliability problems commonly fall in the #P-Complete class of problems. Network reliability can be estimated through the use of a Monte Carlo method by simulating sensor failures over the next δ time units, determining network coverage at the end of the mission, and recording if the mission is successful or not. Repeating this process over a large number of replications allows for an estimation of reliability upon completion. One deterrent of this approach is the unbounded growth of the relative error for highly reliable and highly unreliable networks [13]. Further, improving upon and optimizing a policy through a Monte Carlo method requires a significant computational effort.

3.2.2 Destruction Spectrum

The D-spectrum has been introduced to estimate network reliability [14], and offers several advantages over a traditional Monte Carlo algorithm. First, the D-spectrum yields an efficient representation of the network's reliability but depends only on the system structure. Additionally, while the D-spectrum is also commonly estimated using a Monte Carlo method, it is more efficient than a Monte Carlo algorithm that estimates network reliability [13]. If we consider a network of n sensors subject to failure, the D-spectrum is a probability distribution on the number of failed sensors necessary to cause network failure. Let s_i^n be the probability that in a network of n sensors, the ith sensor failure results in network coverage falling below the requirement α. For the initial WSN that is deployed, every sensors follows an i.i.d. failure distribution F, and network reliability at time t can be estimated by

$$r(t;\alpha,n) = \sum_{i=0}^{n} s_i^n B(i-1;n,F(t)), \tag{10}$$

where $B(i-1;n,F(t))$ is the cumulative binomial probability of no more than $i-1$ success in n trials with probability of success $F(t)$ [15].

Under a M-CBDP sensors will be deployed in the network over a series of missions based on the budget β available leading not only to a variable network size, but also changing the age composition of sensors in the network. As a result, sensors that were deployed in previous missions and remain functioning now have a residual life distribution, denoted T_x where $x > 0$ represents a sensor's age, and fail according to the cdf

$$F_x(t) = \frac{F(x+t) - F(x)}{\bar{F}(x)}. \tag{11}$$

We can use (11) to determine the residual lifetime of a sensor randomly selected in the network, while considering the randomness of its age, as follows. Since network reliability increases along with the number of sensors in the network, the entire budget β will be utilized to deploy new sensors in the network each mission. We can now use the cost constraint, $c_F + c_V x_m \leq \beta$, to determine the maximum number of sensors that can be deployed each mission by

$$\bar{\beta} = \left\lfloor \frac{\beta - c_F}{c_V} \right\rfloor. \tag{12}$$

Every δ time units $\bar{\beta}$ sensors will be pushed into the network, eventually resulting in a stable mix of sensors where the probability distribution on the age k of a randomly selected sensor does not change from on mission to the next. From [16], this probability distribution is described by

$$\rho_k = \frac{\bar{F}(k\delta)}{\sum_{j=0}^{\infty} \bar{F}(j\delta)}, k \in \mathbb{Z}_{\geq 0}. \tag{13}$$

With (11) and (13), the residual lifetime of a sensor in the network, considering the randomness of its age, is now described by the cdf

$$G(t;\delta) = \sum_{k=0}^{\infty} \frac{F(k\delta + t) - F(k\delta)}{\bar{F}(k\delta)} \rho_k, \tag{14a}$$

$$= \frac{\sum_{k=0}^{\infty}[F(k\delta + t) - F(k\delta)]}{\sum_{j=0}^{\infty} \bar{F}(j\delta)}. \tag{14b}$$

The D-spectrum is independent on the failure distribution, but it is impacted by the size of the network. Due to a fixed number of sensors $\bar{\beta}$ deployed in the network each mission and variability in the number of sensor failures, the number of sensors in the WSN will also fluctuate over time. However immediately after new sensors have been deployed, the network consists of

$$n_\beta = \bar{\beta} \sum_{j=0}^{\infty} \bar{F}(j\delta), \tag{15}$$

sensors, on average. The significance of (15) is that we have an expression for the expected size of a WSN in the presence of a M-CBDP with budget β available per mission. Additionally, the remaining life of a sensor randomly selected in the WSN is an i.i.d. random variable with cdf given by (14b).

Finally, to apply the D-spectrum to a M-CBDP we must have knowledge about the system structure, or distribution of sensors in the network. For a fixed budget β available we now know this corresponds to a network with approximately n_β sensors. With an expectation on network size we can now search for the allocation of n_β sensors to each of the subregions to maximize network reliability. Let Y be some policy that determines how the n_β sensors are distributed to each subregion. For example, one policy is to distribute sensors so that each subregion contains approximately the same number of sensors. A policy informs the overall configuration of sensors in the network (i.e., the structure of a network that consist of n_β sensors), in addition to how new sensors are deployed in the network based on the observed state. Policy Y now provides a consistent network structure between missions (that is, after the deployment of sensors each mission the network contains n_β sensors distributed throughout the network in a similar manner), and the D-spectrum can be used to estimate network reliability. Let s_i^Y be the probability the ith sensor failure results in $C(\mathcal{G})$ falling below α when following the M-CBDP Y. Network reliability is estimated by

$$r(\delta;\alpha,\beta,Y) = \sum_{i=0}^{n_\beta} s_i^Y B(i-1; n_\beta, G(\delta;\delta)). \tag{16}$$

Using the network D-spectrum, (16) can be applied to efficiently evaluate network reliability when new sensors are deployed in the network according to a given M-CBDP Y. In the following section we compare the performance of various policies, after which the best policy from those evaluated can be selected.

3.3 Computational Results

In this section we compare the performance of various M-CBDPs for a varying budget, β, and observation interval, δ. To model the failure of sensors, the lifetime of each sensor is distributed according to a Weibull distribution with a shape parameter 1.5 and scale parameter 10. Sensor capabilities are defined based on a common communication radius of $d_1 = 0.075$ and a monitoring radius of $d_2 = 0.075$. The region of interest $\mathcal{R}$ is a $[0,1] \times [0,1]$ square that is patritioned into $n_r = 16$ equal sized regions (i.e., each subregion is of size 0.25×0.25), with a single sink node located centrall in $\mathcal{R}$. The coverage requirement is selected as $\alpha = 0.8$, meaning the WSN must cover 80% of the region to be successful. The fixed cost of deploying sensors is set to $c_F = 100$, with a variable cost of $c_V = 1$.

The first M-CBDP we consider is to evenly distribute sensors to each subregion, denoted policy Y_1. That is, after observing the state of the network new sensors are deployed so that each subregion contains approximately n_β/n_r sensors. As a result, if we observe a subregion that has suffered more failures compared to another, more sensors will be deployed to this subregion. The second myopic policy, Y_2, is to deploy new sensors to a subregion based on a weight w_i assigned to each subregion. Since sensors located closer to the sink node are relied upon more often to route information we may wish to place a larger weight on subregions around the sink in order to deploy a larger number of sensors, providing a level of redundancy and maintaining a communication path in the presence of failures. The weights now influence how new sensors are deployed in the network, where even if we observe a large number of sensors that remain functioning in a subregion it might be advantageous to deploy sensors to this subregion if it is near the sink. For M-CBDP Y_2, the weight of each subregion is inversely proportional to the distance from the sink node to the center of a subregion, and each subregion now contains approximately $(w_i/\sum_{i=1}^{n_r} w_i) \times n_\beta$ sensors. Note that policy Y_1 and Y_2 are not necessarily optimal policies resulting from (9). However they are anticipated to be high quality policies and selected to demonstrate the use of the D-spectrum to estimate the reliability of a CBDP. Future work will be directed on efficient methods to determine an optimal policy beyond an enumeration strategy.

These two M-CBDPs are compared against a simpler time-based deployment policy (TBDP), TB. In policy TB, rather than deploy sensors based on a budget available, sensors are deployed to reach a constant network size. Additionally, only the number of sensors functioning in the network is observed and sensors are then randomly deployed throughout the entire region, instead of specifying the subregion a sensor is deployed in. A TBDP is explained in more detail in [17]. Results for the two M-CBDPs along with the TBDP are provided in Table 2 where the values under each policy correspond to the resulting network reliability estimated using (16), given β and δ.

Table 2: Network Reliability for Various M-CBDPs

β	δ	Y_1	Y_2	TB	β	δ	Y_1	Y_2	TB
278	2.5	0.9998	0.9999	0.9998	388	5.7	0.8004	0.8123	0.7359
364	5.0	0.9499	0.9598	0.9227	383	5.7	0.7503	0.7667	0.6820
353	5.0	0.8999	0.9136	0.8557	594	8.2	0.7003	0.7200	0.6542
353	5.1	0.8507	0.8719	0.7981	438	6.5	0.7002	0.7199	0.6357

In each of the test instances, M-CBDP Y_2 results in the largest reliability, followed by M-CBDP Y_1, and finally the TBDP TB. One of the primary differences between policy Y_1 and Y_2 with TB is that in Y_1 and Y_2 we are able to observe the state of the network and determine how new sensors are deployed in the region (i.e., which subregion sensors are deployed in). This is a significant improvement over policy TB, particularly as the time between network observation increases. For example, in the instance with $(\beta, \delta) = (388, 5.7)$, this results in an improvement in network reliability from 0.7359 for the TBDP to 0.8004 for M-CBDP Y_1. By weighting each subregion and influencing the M-CBDP through this method (policy Y_2), network reliability is improved further.

The test instances also help illustrate the impact of β and δ on each policy. For example, consider the $(353, 5.0)$ instance and the $(353, 5.1)$ instance. The observation interval in the latter instance is slightly larger, but this results in a drop in network reliability from 0.9136 to 0.8719 for policy Y_2, with a similar impact on policy Y_1. In the following set of test instances, $(388, 5.7)$ and $(383, 5.7)$, the observation interval is the same but the mission budget has slightly decreased. With a variable cost $c_V = 1$, this corresponds to five fewer sensors available to deploy per mission in the second scenario. However, this again results in a drop in network reliability from 0.8123 to 0.7667 for policy Y_2.

Finally, because we are using an estimate of the D-spectrum for an approximate size of the network to estimate reliability under a CBDP, we are interested in how accurate this estimate is compared to a traditional Monte Carlo simulation. Although a Monte Carlo simulation is more computationally expensive, it provides the ability to model the fluctuation in network size and in the age of sensors over time. For a Monte Carlo simulation of 10000 replications on the $(353, 5.0)$ instance, the resulting reliability estimate is 0.8993 and 0.9151 for policy Y_1 and Y_2, respectively. A Monte Carlo simulation for the remaining test instances yields a similar performance comparison, demonstrating the suitability of the D-spectrum to estimate reliability of a M-CBDP.

3.4 Conclusion

To maintain a WSN over a prolonged period of time, new sensors must be deployed in the network to re-establish network coverage and communication capabilities. Towards this goal, we have discussed a myopic condition-based sensor deployment problem in which the network is observed prior to a decision on how new sensors are deployed in the network. We have also demonstrated how the network D-spectrum can be used to estimate network reliability as new sensors are deployed, and compare the performance of different sensor deployment policies. With this insight to a M-CBDP, future work is focused on a model that considers the impact on future mission reliability as well.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] S. Debnath, A. K. Singh, and A. Hossain, "A comprehensive survey of coverage problem and efficient sensor deployment strategies in wireless sensor networks," *Indian Journal of Science and Technology*, vol. 9, no. 45, pp. 1–6, 2016.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, 2002.

[4] N. Li and J. C. Hou, "Flss: a fault-tolerant topology control algorithm for wireless networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*. ACM, 2004, pp. 275–286.

[5] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. New York, NY: ACM, 2003, pp. 28–39.

[6] P. K. Sahoo and J.-P. Sheu, "Limited mobility coverage and connectivity maintenance protocols for wireless sensor networks," *Computer Networks*, vol. 55, no. 13, pp. 2856–2872, 2011.

[7] S. Ganeriwal, A. Kansal, and M. B. Srivastava, "Self aware actuation for fault repair in sensor networks," in *Proceedings IEEE International Conference on Robotics and Automation, 2004.*, vol. 5. IEEE, 2004, pp. 5244–5249.

[8] H. M. Almasaeid and A. E. Kamal, "On the minimum k-connectivity repair in wireless sensor networks," in *2009 IEEE International Conference on Communications*. IEEE, 2009, pp. 1–5.

[9] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Networks*, vol. 14, no. 3, pp. 347–355, 2008.

[10] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.

[11] M. O. Ball, "Computational complexity of network reliability analysis: An overview," *IEEE Transactions on Reliability*, vol. 35, no. 3, pp. 230–239, Aug 1986.

[12] S. Misra, S. R. Mohan, and R. Choudhuri, "A probabilistic approach to minimize the conjunctive costs of node replacement and performance loss in the management of wireless sensor networks," *IEEE Transactions on Network and Service Management*, vol. 7, no. 2, pp. 107–117, 2010.

[13] Y. Shpungin, "Networks with unreliable nodes and edges: Monte carlo lifetime estimation," *Applied Mathematics and Computer Science*, vol. 27, pp. 168–173, 2007.

[14] F. J. Samaniego, "On closure of the ifr class under formation of coherent systems," *IEEE Transactions on Reliability*, vol. R-34, no. 1, pp. 69–72, April 1985.

[15] J. Navarro, F. J. Samaniego, N. Balakrishnan, and D. Bhattacharya, "On the application and extension of system signatures in engineering reliability," *Naval Research Logistics*, vol. 55, no. 4, pp. 313–327, 2008.

[16] M. Finkelstein and J. Vaupel, "On random age and remaining lifetime for populations of items," *Applied Stochastic Models in Business and Industry*, vol. 31, no. 5, pp. 681–689, 2015.

[17] N. T. Boardman and K. M. Sullivan, "Time-based node deployment policies for reliable wireless sensor networks," *IEEE Transactions on Reliability*, pp. 1–14, 2021.

4 Approximate Dynamic Programming for Condition-Based Node Deployment in a Wireless Sensor Network

Abstract

The flexibility of deployment strategies combined with the low cost of individual sensor nodes allow wireless sensor networks (WSNs) to be integrated into a variety of applications. It is important that the WSN function over a prolonged period of time, particularly as sensors consume a finite power supply and begin to fail. Extending network lifetime is possible by deploying new sensors in the network, and is commonly concerned with a single stage deployment to restore a network coverage and/or communication measure. In this work we focus on condition-based deployment policies (CBDPs) in which sensors are deployed over a series of missions. The main contribution is a Markov decision process (MDP) model to maintain a reliable WSN with respect to region coverage. Due to the resulting high dimensional state and action space, we explore approximate dynamic programming (ADP) methodology in the search for high quality CBDPs. Our model is one of the few related to maintenance through the repeated deployment of new sensor nodes, and one of the first ADP applications for the maintenance of a complex WSN. Additionally, our methodology incorporates a destruction spectrum (D-spectrum) reliability estimate, addressing the complexity present in both a dynamic network topology and dynamic age composition of sensors. While the D-spectrum has received significant attention with respect to network reliability, the application and utility in a maintenance setting has not been widely explored. We conclude with a discussion on CBDPs in a range of test instances, and comparisons with alternative deployment strategies.

4.1 Introduction

Through the cooperative effort of individual sensor nodes, a wireless sensor network (WSN) can be deployed to monitor and report data on an event of interest in a desired region. WSNs can also be left unattended once they are deployed, allowing them to be integrated into a wide range of applications. In environmental settings WSNs can be valuable to monitor a forest providing early detection of forest fires, or to monitor a coastline and warn about potential flooding [1]. WSNs have additionally been deployed to observe animals and their behavior in a natural habitat over a period of time with minimal disruption [2]. In commercial applications, WSNs can be utilized to track inventory or for temperature/climate control in buildings and warehouses [3]. Sensors have also been integrated into military and healthcare applications [4], illustrating the flexibility WSNs offer.

While the area a single sensor is able to monitor can be relatively small, sensors are able to communicate with neighboring sensors to route information through the network. By sufficiently distributing sensors throughout a region of interest, the WSN is able to monitor a much larger region. In some environments (e.g., buildings) this is possible by placing sensors at specific locations, and may require fewer sensors to monitor the region effectively [4]. In other scenarios (e.g., dense forest, mountainside) where the construction of a specific network topology is more difficult, a WSN can be quickly established by randomly deploying sensors over the target area [5]. This may require a larger number of sensors compared to a controlled deployment, but the low cost nature of sensor nodes facilitate large WSNs that are not cost prohibitive [3]. The randomness in sensor deployment can also be influenced by the density with which sensors are deployed throughout the network, for example by deploying a sensor over a desired subregion in the network [6]. Further contributing to the feasibility of a random deployment, sensors require very little infrastructure (e.g., wires, cables) to operate. Each sensor contains components necessary for sensing and sending/receiving data, as well as an individual power supply such as a battery [7]. Sensors consume this finite power supply over the course of network operation until there is no power remaining and the sensor ceases to function properly. As an increasing number of sensors reach this failure stage, the overall network coverage and connectivity also start to decline.

One method to delay the impact of sensor failures on network lifetime is through the use of sleep/wake cycles [8–10]. This allows redundant sensors to remain in an energy conserving state until they are required to turn on and assist network functions. Dynamic power management methods can also be implemented, which adjust the transmission power of individual sensors to minimize the energy required while ensuring sensors can still communicate with one another [11, 12]. For the deployment of sensors at a single point in time topology control methods such as these can be an effective means to extend network lifetime. However the issue of a finite power supply and a reduction in the number of functional sensors over time is still encountered. To contribute toward the long-term operation of a WSN we must consider additional methods to address the impact of sensor failures. In applications where the WSN is easily accessible it may be possible to replace or recharge the battery of failed sensors. Several different node replacement policies are examined in [13], where there are a limited number of replacement sensors available and the decision to replace a sensor or not is made immediately after observing a failure. However there are many environments in which it is not practical to access failed nodes individually [14]. Another option to improve network status, viable through the low cost of sensor nodes combined with the lack of physical connection between sensors, is to deploy new sensors (at new locations) in the WSN.

An additional emphasis is commonly placed on determining a minimal number of relay sensors, more powerful (e.g., larger communication radius) but more costly sensors, directed toward restoring sensor connectivity. This is the focus in [15] and [16], which explore algorithms that determine the fewest number of relay sensors and their placement in the network. The relay sensor placement problem restores a communication ability in the network, but there is the possibility the next sensor failure immediately causes the network to become disconnected again. One attempt to address this issue is to deploy sensors such that a level of redundancy or k-connectivity is present in the network [17, 18]. The resulting network now has capability restored, in additional to a network intended to be robust to sensor failures. One of the limitations in the previous models is that they are commonly concerned with the deployment of sensors at a single point in time. The age of sensors in the network is often not considered which is important as sensors that remain functioning in the network are 'older' compared to brand new sensors, and their residual life is likely smaller. As a result, they are more likely to fail and there is the possibility the network becomes disconnected again in the near future, requiring the deployment of even more relay sensors. Since the long-term maintenance of a WSN will require the deployment of sensors at several different stages, a deployment policy can be improved by considering the residual life distribution of sensors remaining in the network and the frequency with which sensors must be deployed. The newly deployed sensors are also not required to be relay sensors; the deployment of a larger number of new, identical sensors is possible and may help manage the cost over a series of deployment actions.

When a WSN is maintained in this manner, a decision must be made on how many new sensors are deployed and where in the network to improve the ability of sensors to communicate with each other and restore coverage in portions of the network that has been lost. Limiting the ability to deploy sensors is a finite resource, for example a budget or limited time window to access the network and deploy new sensors. This maintenance decision relates closely to the selective maintenance problem, in which an action must be selected from the many set of feasible actions available. A mathematical formulation of the selective maintenance problem in a series-parallel system is discussed in [19], where models are presented that maximize system reliability subject to constraints on cost and maintenance time available, or minimize cost (time) subject to a constraint on the time (cost) and minimum system reliability requirement. In [20] the model is expanded to consider multiple maintenance actions (e.g., minimally repair failed components, replace failed components, replace functioning components), and model the lifetime of an individual component with a Weibull failure distribution. In both [19] and [20] the maintenance decision is based on maximizing or minimizing the objective for the next mission (i.e., until the next maintenance action). Since the system is likely maintained over a series of missions, a maintenance policy can be improved by considering the impact of a decision on future missions as well. This problem is first explored in [21] through a Markov decision process (MDP) model for a small series-parallel system, and later in [22] by applying approximate dynamic programming (ADP) methodology to solve for a maintenance policy in a system comprised of a larger number of subsystems and components.

Similar to a series-parallel system subject to component failure, WSNs are stochastic systems and allow a wide range of opportunities to incorporate MDP methodology. A survey of MDP models applied to various problems involving WSNs is provided in [23]. Topics related to resource and power optimization (e.g., sleep/wake cycles, battery recharge policies), data exchange and topology formulation (e.g., transmission radius management, data aggregation), and sensing coverage and object detection (e.g., tracking a mobile object) are a few of the areas MDP models have been applied to. With respect to deploying new sensors in a WSN, [24] presents an MDP model that minimizes the cost of replacing failed nodes. One of the limitations in the model is it does not consider network topology, and instead assumes that all failed nodes equally effect the performance of the WSN. Large scale networks commonly feature redundant sensors and can withstand the first few failures with relatively little impact, but as the number of failures increases it is more likely a 'critical' sensor failure (e.g., the only sensor that monitors a target or connects two portions of the WSN) results in a large drop in network capability.

Compared to the selective maintenance problems discussed in [19–22] that maximize system reliability, research related to WSNs has focused primarily on maximizing lifetime for a given network, replacing a small number failed nodes to maintain network capability, or deploying the fewest number of new sensors to restore the network to a functioning status. WSNs typically lack the well defined structure of a series-parallel system which complicate the estimation of network reliability. As a result, network reliability problems commonly fall in the #P-Compete class of problems [25], and can be difficult to solve exactly. This is further complicated in a maintenance setting in that the number of sensors in the network is changing over time, and the collection of sensors is heterogeneous. That is, sensors have different residual life distributions due to a variation in sensor age, and the age distribution of sensors is also changing over time.

In addition to the complexity present in network reliability, one of the difficulties commonly encountered in MDPs is the high dimensional variables that model the state space, decision space, and/or outcome space. Known as the "three curses of dimensionality," solving even moderately sized versions of a problem can quickly become intractable. Instead of solving the problem exactly, ADP provides methodology to select a decision based on an approximation of the future value (e.g., cost, reward, etc.) associated with an action. A brief introduction to ADP is outlined in [26], a summary of common techniques in [27], and a detailed discussion is provided in [28].

In this work we focus on the selective maintenance of a WSN where new sensors are deployed over a series of several maintenance actions. Prior to the deployment of new sensors, the WSN is observed to provide information related to the current state and inform a decision on how new sensors are deployed in the network. Constraining the set of feasible actions is a limited budget available over a finite planning horizon. While the deployment of new sensors is immediately concerned with restoring/maintaining a level of network coverage and connectivity, the deployment of sensors at a current point in time may also influence how sensors are deployed in future time periods. One of the main contributions of this work is an MDP model to examine an optimal condition-based deployment policy (CBDP) to maintain a WSN for region coverage. Due to the resulting high dimensional state and maintenance (i.e., sensor deployment) decision space, we explore ADP methodology to address the complexity present in a repeated sensor deployment setting. While MDP models have been widely applied to WSNs, our model is one of the few related to maintenance through the repeated deployment of new sensor nodes, and one of the first ADP applications for the maintenance of a complex WSN.

Additionally, our model focuses on maximizing a measure of network reliability. Due to the complexity encountered in network reliability problems, approximation methods such as a Monte Carlo simulation [29] or bounding network reliability [30, 31] are commonly utilized. A Monte Carlo simulation can be used to evaluate the performance of a given policy, but improving upon and optimizing a policy is a more difficult task. Network reliability can also be estimated utilizing the destruction spectrum (D-spectrum), introduced in [32], and is a function of the network structure, or the number of sensors and their distribution (i.e., locations) throughout the network. The D-spectrum has received significant attention in network reliability literature, but its application in a maintenance setting is still emerging. A model for a node replacement policy when a network is subject to external shocks causing a sensor failure with equal probability is provided in [33], where the decision is when (i.e., after how many shocks) to replace failed nodes. A time-based deployment policy (TBDP) for a WSN is explored in [34] where the network is restored to a fixed size at periodic time intervals, allowing the D-spectrum to be applied in evaluating a wide range of policies. Closely related to a TBDP is one in which a fixed number of sensors are deployed in the network at constant time intervals. This now results in a varying network size, but the D-spectrum remains valuable in this problem [35]. We discuss how the D-spectrum can be adapted into a model to estimate reliability in the presence of a CBDP, which must address the complications concerning both a dynamic network topology and a dynamic age composition of sensors.

The remainder of this work is organized as follows. Section 4.2 summarizes the modeling of a WSN and outlines the progression of network observation followed by a deployment decision over a period of time. This sequence of events informs a discussion on an MDP model for the condition-based sensor deployment problem. Section 4.3 discusses our ADP methodology to determine an optimal CBDP and addresses how the the D-spectrum can be incoporated to estimate reliability based on an observed state and action. Section 4.4 presents numerical results for a range of test instances and compares optimal ADP policies to alternative deployment strategies. Finally, Section 4.5 concludes this article, and provides a few directions for future research.

4.2 Problem Description and Model

In this section we discuss a condition-based sensor deployment MDP model in which a limited budget is available to deploy additional sensors in the network. Consider a WSN deployed over some region of interest, partitioned into a number of smaller subregions represented by the indexed set $\mathcal{R} = \{1, 2, \ldots, n_r\}$. The WSN, represented by $\mathcal{G}$, is comprised of a collection of sensor nodes and a single sink node located somewhere in the region. Sensors in the network are responsible for communicating with neighboring sensor nodes to route information through the network, with a desired destination at the sink node. In addition to a communication capability, sensors are tasked with monitoring the surrounding area and desired target locations in the region. These two primary sensor capabilities are defined by a communication radius $d_1 > 0$, and a monitoring radius $d_2 > 0$. For a target to be covered in the network it must not only be within the monitoring radius of a functioning sensor; there must also be a communication path from the monitoring sensor back to the sink node. The ability of sensors to communicate with one another declines over time as a result of sensor failures, which also impacts the collection of targets covered. The lifetime of an individual sensor is modeled by a survival function $\bar{F}(t) = 1 - F(t)$, where $F(t)$ represents the cumulative distribution function (cdf) of sensor lifetime and is assumed to be identically distributed for all sensors. At time $t \geq 0$, the WSN $\mathcal{G}$ is represented by $\mathcal{G}(t)$ and consists

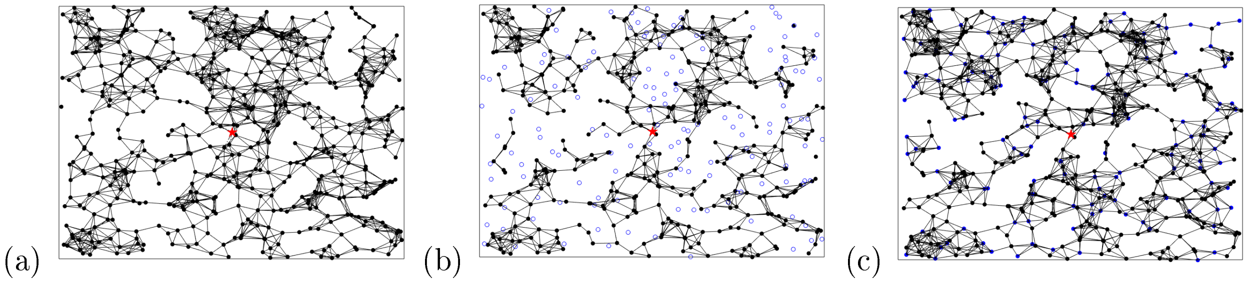

Figure 9: (a) Initial WSN with sink node ($\star$) and functioning sensor nodes ($\bullet$) ; (b) WSN with failed sensors ($\circ$) ; (c) WSN with newly deployed sensors ($\bullet$).

of sensors that remain functioning at time t. The proportion of targets covered, or WSN coverage, is denoted $C(\mathcal{G}(t))$ and informs the condition of the network.

An example of the WSN evolution over time is illustrated in Figure 9. In Figure 9(a) the WSN contains a large number of sensors and covers a significant portion of the region. Over time sensors fail and can dramatically impact network performance, as illustrated in Figure 9(b). To prevent a further drop in coverage and restore network capability, new sensors are deployed in the network, demonstrated in Figure 9(c). New sensors can be deployed in the network with an objective to improve the ability of sensors to communicate with one another, in addition to re-establishing coverage in portions of the network that were severely impacted by failures.

The desire of deploying new sensors in the WSN is to enable the region of interest to be monitored over a sequence of missions $\{0, 1, \ldots, M-1\}$. Each mission is of equal duration δ, and mission m corresponds to the duration of time between $m\delta$ and $(m+1)\delta$. Additionally, it is assumed that the starting number of sensors deployed in the network at time $t = 0$ is given. The first redeployment action therefore corresponds to mission 1 at time $t = \delta$. At the beginning of mission 1, and each subsequent mission, the network is observed and a decision is then made on how new sensors are deployed in the network. In our discussion throughout we adopt the convention that network observation and the deployment of any new sensors always occur at the beginning of a mission. Since the end of mission $m-1$ corresponds to the beginning of mission m, an equivalent statement is that the network is observed at the end of mission $m-1$, the deployment of new sensors occurs, and then mission m starts. For consistency purposes and ease of state variable and decision variable definitions introduced later, we always refer to both actions occurring at the beginning of a mission.

To avoid the effort/cost involved in deploying a sensor to a specific location (or replacing a failed sensor), the subregion a sensor is deployed in is selected and the sensor is then randomly deployed within the subregion. This is similar to varying the density with which sensors are deployed throughout the network [36]. The decision is now how many sensors are deployed, and in which subregion of the network. When new sensors are deployed in the WSN, a fixed cost c_F is incurred if at least one sensor is deployed in addition to a variable cost c_V for each sensor deployed. The fixed cost plus variable cost model relates to the hardware plus non-hardware model discussed in [24], and is also used in a related work investigating time-based redeployment policies [34]. It is assumed that all sensors deployed in the network are homogeneous, in the sense that all sensor capabilities are identical and sensors follow and independent and identically distributed (i.i.d.) failure distribution, F.

Since new sensors are deployed in the network over a sequence of missions, the collection of sensors is heterogeneous in the sense that sensors have different ages, and therefore different residual life distributions. Let k be the age of a sensor in the network, where sensors are deployed with initial age $k = 0$. The age of a sensor therefore corresponds to how many missions the sensor has survived. Define $\mathcal{K} = \{0, 1, \ldots, K\}$ as the set of all possible ages, where K is some upper bound on the age of a sensor in the network.

The state space consists of two main components, the first of which is the observed distribution of sensors in the network and is defined as

$$N_m = (N_{mik})_{i\in\mathcal{R},k\in\mathcal{K}} \equiv (N_{m10}, N_{m11}, \ldots, N_{m1K}, N_{m20}, \ldots, N_{mn_rK}), \tag{17}$$

where N_{mik} denotes the number of functioning sensors with age $k \in \mathcal{K}$ in subregion $i \in \mathcal{R}$ at the beginning of mission m. The total number of functioning sensors in the network is denoted by $\bar{N}_m = \sum_{i\in\mathcal{R}} \sum_{k\in\mathcal{K}} N_{mik}$. The second component of the state space is the budget available to deploy sensors during mission m (and all future missions), denoted B_m. Combining these two components, the state of the system at the beginning of mission m is defined by $S_m = (N_m, B_m) \in \mathcal{S}$, where $\mathcal{S}$ is the set of all possible states.

After observing the state of the network, a decision must be made on how new sensors are deployed. Let x_{mi} denote the number of sensors deployed in subregion $i \in \mathcal{R}$ at the beginning of mission m, and $\bar{x}_m = \sum_{i\in\mathcal{R}} x_{mi}$ be the total number of sensors deployed. The resulting cost from observing state S_m and implementing action x_m is denoted $C_m(S_m, x_m)$, where

$$C_m(S_m, x_m) = \begin{cases} c_F + c_V\bar{x}_m, & \text{if } \bar{x}_m > 0, \\ \\ 0, & \text{othwerwise.} \end{cases} \tag{18}$$

The transition probability functions can now be used to characterize how the system evolves from one state to another. First, note that an individual sensor with age k survives the current mission with probability

$$p_k = \frac{\bar{F}((k+1)\delta)}{\bar{F}(k\delta)}. \tag{19}$$

Using the survival probability for an individual sensor, the transition probability for the number of sensors with age k in subregion i is determined by

$$\Pr(N_{m+1,i,k}|N_{mik-1}, x_m) = \begin{cases} b(N_{m+1,i,1}; x_{mi}, p_{k-1}), & \text{if } k = 1 \text{ and } 0 \leq N_{m+1,i,k} \leq x_{mi}, \\ b(N_{m+1,i,k}; N_{mik-1}, p_{k-1}), & \text{if } k > 1 \text{ and } 0 \leq N_{m+1,i,k} \leq N_{mik-1}. \end{cases} \tag{20}$$

where $b(n; x, p)$ is the binomial probability of n successes in x trials with probability of success p. The overall transition probability given maintenance action x_m can now be determined by

$$\Pr(N_{m+1}|N_m, x_m) = \prod_{i \in \mathcal{R}} \prod_{k \in \mathcal{K}} \Pr(N_{m+1,i,k}|N_{mik-1}, x_m). \tag{21}$$

The second component of the state variable is the budget, which transitions based on the corresponding cost of the action implemented,

$$B_{m+1} = B_m - C_m(S_m, x_m). \tag{22}$$

The state transition function is defined as $S_{m+1} = S^M(S_m, x_m, W_{m+1})$, where W_{m+1} represents information on sensor failures that occur during mission m.

Given a starting budget B_0, the objective is to deploy sensors in the network to maximize the expected number of successful missions. For a given coverage requirement α, an individual mission is successful if WSN coverage over the duration of the mission remains above this requirement. Network reliability is also defined with respect to α, and is defined as the probability the coverage requirement is satisfied over the mission duration. From an observed network state S_m and implementing action x_m, the resulting network reliability is denoted $R_m(S_m, x_m)$. Let $X_m^\pi(S_m)$ be a policy that determines the sensor deployment action (how many sensors that are deployed and in which subregions) for each state $S_m \in \mathcal{S}$. For a given number of missions M, the objective is

$$\max_{\pi \in \Pi} \mathbb{E}^\pi \Big\{ \sum_{m=0}^{M-1} R_m(S_m, X_m^\pi(S_m)) \Big\}. \tag{23}$$

Constraining a decision each mission is first the budget available, B_m, to deploy sensors in the network. Additionally, there may be some desired minimum reliability (i.e., probability of mission success), ϕ, that each mission achieve. This constraint is intended to prevent the scenario where network reliability is completely sacrificed (i.e., unacceptably low reliability and almost certain network failure) one mission, while the reliability of a later mission is near one. Finally, there may exist an upper limit on the number of sensors allowed in the network, n^{max}, to prevent the region from becoming saturated with sensors at any given time. Overall the set of feasible actions, $\mathcal{X}_{S_m}$, during mission m is therefore defined by

$$\mathcal{X}_{S_m} = \Big\{ x_m : C_m(S_m, x_m) \le B_m, R_m(S_m, x_m) \ge \phi, \bar{N}_m + \bar{x}_m \le n^{max} \Big\}. \tag{24}$$

One of the complicating aspects in determining the set of feasible actions is the reliability requirement an action must satisfy. As previously mentioned, network reliability problems commonly fall in the #P-Compete class of problems, and therefore determining the exact set of feasible actions as defined by (24) is not a trivial task. Later on we address this difficulty by outlining an efficient method to estimate network reliability and instead apply the constraint to the estimated reliability of an action, $\hat{R}_m(S_m, x_m)$. In doing so the set of feasible actions is now approximated as well, and it is possible our approximation includes actions that are not feasible to (24). That is, the estimated reliability of an action may satisfy the constraint and therefore appear in our approximated action set, but the true value might be below the requirement. However, this should only occur for a small number of actions, and the actions *are* feasible to the original problem with only a cost constraint.

The value function, $V_m(S_m)$, is defined as the maximum number of successful missions remaining among missions $m, m+1, \ldots, M-1$ if the system is in state S_m at the beginning of mission m. To determine an optimal policy to (23) we must find a solution to Bellman's equation,

$$V_m(S_m) = \max_{x_m \in \mathcal{X}_{S_m}} \Big\{ R_m(S_m, x_m) + \mathbb{E}\big[V_{m+1}(S_{m+1}) \big| S_m, x_m\big] \Big\}. \tag{25}$$

4.3 ADP Formulation

The previous section provides an initial MDP model for the condition-based sensor deployment problem over a sequence of M missions. Common to many dynamic programming problems, this model suffers from the three curses of dimensionality [28]. The large size of the state space can be illustrated by examining the distribution of sensors in the network. For a network containing i sensors, these sensors can be allocated to different regions of the network $\binom{n_r+i-1}{i}$ different ways. Due to sensor failures and the deployment of new sensors, the total number of sensors in the network also varies between 0 and n^{max}. As a result, the size of the state space considering only the distribution of sensors in the network is $\sum_{i=0}^{n^{max}} \binom{n_r+i-1}{i}$. Note that this does not include any information about the age composition of sensors, which further complicates the size of the state space. The remaining budget is also a factor, and can bounded between 0 and B_0. Assuming integer values of c_F and c_V then the budget for mission m can also assume integer values between 0 and B_0, and the size of the state space can be bounded by $B_0 \sum_{i=0}^{n^{max}} \binom{n_r+i-1}{i}$ for a single mission. The large action space (i.e., deciding how new sensors are deployed in different subregions) and outcome space (i.e., observing sensor failures) are additional components that limit exact algorithms to be applied for only small problem instances.

For large scale WSNs of interest, ADP can be applied to the condition-based sensor deployment problem. First, the optimality equations can be reformulated around the post-decision state variable, S_m^x, which is the state at the beginning of mission m immediately after new sensors have been deployed in the network. Similarly, the number of sensors functioning in each subregion immediately after new sensors have been deployed and the total number of sensors in the network are represented by N_m^x and $\bar{N}_m^x$, respectively. Let $V_m^x(S_m^x)$ denote the value of being in the post-decision state S_m^x, and is defined as the maximum number of successful missions among missions $m+1, m+2, \ldots, M-1$ given the post-decision state variable S_m^x. The relationship between V_m^x and V_m can be expressed by

$$V_{m-1}^x(S_{m-1}^x) = \mathbb{E}[V_m(S_m)|S_{m-1}^x], \tag{26}$$

where

$$V_m(S_m) = \max_{x_m \in \mathcal{X}_{S_m}} \Big\{ R_m(S_m, x_m) + V_m^x(S_m^x) \Big\}. \tag{27}$$

Substituting (27) into (26) we obtain the optimality equations around the post-decision state variable

$$V_{m-1}^x(S_{m-1}^x) = \mathbb{E}\Big\{ \max_{x_m \in \mathcal{X}_{S_m}} \big(R_m(S_m, x_m) + V_m^x(S_m^x) \big) | S_{m-1}^x \Big\}. \tag{28}$$

One of the advantages of utilizing the post-decision state variable is the expectation is now outside of the maximization problem. The resulting maximization problem in (28) is less complicated than the original formulation in (25), but still requires an evaluation of network reliability. The D-spectrum has been utilized to estimate network reliability, and offers several advantages over a traditional Monte Carlo simulation. Notably, in the presence of i.i.d. sensor failures the D-spectrum is only a function of the network structure, and does not depend on the failure distribution of sensors in the network [37]. While it is possible to compute the D-spectrum of a network exactly it is more common to use an approximation method, particularly when applied to a large, complex WSN. One such approximation method relies on the use of a Monte Carlo method, however a Monte Carlo estimation of the D-spectrum is more efficient compared to a traditional Monte Carlo simulation that directly estimates network reliability [29]. The lower computational effort required in estimating the D-spectrum, algorithms of which are outlined in [34] and [38], becomes significant when reliability estimation is embedded in an optimization problem and may need to be repeated over a large number of replications.

4.3.1 Destruction Spectrum Reliability Estimation

In a network of sensors subject to failure, the D-spectrum is a probability distribution on the number of failed sensors that result in network failure. From information available in the post-decision state variable we can apply the network D-spectrum to estimate reliability, but must first define a number of state aggregation functions. Let $\mathcal{S}^{(a)}$ be the state space at the ath level of aggregation, where the aggregation function A^a maps the original state space $\mathcal{S}$ to $\mathcal{S}^{(a)}$. Define A^1 as the function that aggregates over the age composition of sensors in a subregion, resulting in the number of sensors in each subregion, $N_m^{(1)} = (N_{m1}^{(1)}, N_{m2}^{(1)}, \ldots, N_{mn_r}^{(1)})$. The second aggregation function, A^2, aggregates over the subregions in the network, resulting in the number of sensors with a given age, $N_m^{(2)} = (N_{m0}^{(2)}, N_{m1}^{(2)}, \ldots, N_{mK}^{(2)})$.

Applying the first aggregation function to the post-decision state variable, we can determine the number of sensors functioning in each subregion. This is significant as it now provides information on the resulting network structure, and we can estimate the corresponding D-spectrum (i.e., the D-spectrum for a network with $N_{mi}^{x,(1)}, i \in \mathcal{R}$ sensors randomly located in each subregion). The D-spectrum estimate is denoted $\hat{s}_{\alpha,i}^{\bar{N}_m^x}$, and is the probability the ith sensor failure results in network coverage falling below the requirement α in a network of $\bar{N}_m^x$ sensors. From the second aggregation function we can determine the probability of randomly selecting a sensor with age k in the network by

$$\tilde{\rho}_k = \frac{N_{mk}^{x,(2)}}{\bar{N}_m^x}, \;\; k \in \mathbb{Z}_{\geq 0}. \tag{29}$$

With (29), the residual life distribution for a sensor randomly selected in the network is now given by the cdf

$$\begin{aligned}
\tilde{G}(t;\delta) &= \sum_{k=0}^{\infty} \frac{F(k\delta + t) - F(k\delta)}{\bar{F}(k\delta)} \tilde{\rho}_k, && \text{(30a)} \\
&= \sum_{k=0}^{\infty} \frac{F(k\delta + t) - F(k\delta)}{\bar{F}(k\delta)} \frac{N_{mk}^{x,(2)}}{\bar{N}_m^x}. && \text{(30b)}
\end{aligned}$$

From the D-spectrum estimate and residual life distribution in (30b), network reliability over the next mission, given the observed state S_m and action x_m can be estimated by

$$\hat{R}_m(S_m, x_m) = \sum_{i=0}^{\bar{N}_m^x} \hat{s}_{\alpha,i}^{\bar{N}_m^x} B(i-1; \bar{N}_m^x, \tilde{G}(\delta;\delta)), \tag{31}$$

where $B(i-1; \bar{N}_m^x, \tilde{G}(\delta;\delta))$ is the cumulative binomial probability distribution of no more than $i-1$ successes in $\bar{N}_m^x$ trials with probability of success $\tilde{G}(\delta;\delta)$ [37]. The reliability D-spectrum estimate resulting from (31) is also attractive in that the reliability estimate has bounded relative error [38].

One of the limitations of the proposed approach to reliability estimation is that it uses the stable residual life distribution derived in (30b), which relies on a probability distribution of sensor ages aggregated over the entire network. Since we observe information on the age distribution of sensors within a subregion, it is reasonable to question why this level of detail is not retained and incorporated in our estimation method. That is, the residual life distribution can be subregion dependent and more accurately reflect the state of the network. The disadvantage of this approach is it now requires an application of the multi-dimensional D-spectrum [39] which is more complicated to estimate. Additionally, empirical testing comparing the reliability estimate of the aggregated approach to a Monte Carlo simulation indicates that any improvement achieved by the multi-dimensional D-spectrum estimate will be minor. Similarly, the aggregated approach requires a single dimensional D-spectrum in the reliability estimate given by (31), which provides the opportunity to avoid a significant source of computational effort as discussed later in Section 4.3.3.

4.3.2 Value Function Approximation

Another powerful technique in ADP methodology to address the large state space is through approximating the value function. In this work we approach the value function approximation through the use of the previously defined aggregation functions and lookup tables. This is based on the observation that the age composition of sensors in the network and the distribution of sensors contribute greatly to the size of the state space. The former is necessary to estimate the stable residual life distribution while the latter is necessary to estimate the destruction spectrum, both of which are required to estimate reliability of the current mission. It is reasonable to expect that while both of these components will impact future missions as well, the primary factor impacting future missions can be summarized by the size of the network. Therefore, we can aggregate over the age composition and distribution of sensors to determine the total number of sensors in the network, $\bar{N}_m$, to estimate the value function.

Additionally, the starting budget B_0 influences the size of the state space and impacts the ability to deploy new sensors in the network. Assuming the variable cost of deploying additional sensors is relatively small (particularly compared to the fixed cost), deploying one or two additional sensors has a minor impact on the budget remaining. It is also reasonable to assume that the impact of deploying one or two additional sensors has a minor increase to the overall value function, particularly when compared to the impact of deploying 15 to 20 additional sensors. As a result we can aggregate the budget into different intervals corresponding to a range of values that result in a similar state value. If the budget is aggregated into intervals of size d, there are now $\bar{B}_0 = \lceil \frac{B_0}{d} \rceil$ different budget states.

The approximate value function for a given post-decision state S_m^x is denoted $\bar{V}_m(S_m^x)$, and with an aggregated state space size of approximately $\bar{B}_0 \times n^{max}$ is significantly smaller than the original state space.

4.3.3 Determining An Optimal Action

The primary question that remains is addressing how the maximization problem in (28) is solved for the optimal value and corresponding action. From the observed state S_m, we can first determine an upper bound on the number of sensors that can be deployed by $\tilde{n} = n^{max} - \bar{N}_m$ (assuming the budget does not limit us first). This results in a range of $(0, \tilde{n})$ to search for the optimal deployment of new sensors and the regions they are deployed in.

However, deciding how sensors are deployed to various subregions remains a significant task. We can simplify this problem through the assumption that the network size is also indicative of the network structure. That is, if the network contains some number of sensors, and more accurately the post-decision state network consisting of $\bar{N}_m^x$ sensors, these sensors will be distributed throughout the network in roughly the same manner regardless of the age composition of the sensors and the current mission. The implication of this assumption is that our decision transforms from selecting how many sensors to deploy and in which subregions they are deployed in, to only selecting how many sensors are deployed in the network. Once the number of sensors to deploy, $\bar{x}_m$, has been selected, the observed state N_m will determine how many sensors are deployed in each subregion (i.e., x_m) to achieve the desired network structure specified by $\bar{N}_m^x$. This assumption is primarily limiting for a scenario in which a large number of sensors in a subregion all survive for a prolonged period of time, and therefore are more likely to fail during the same mission resulting in a sharp drop in region coverage. However this scenario occurs with very small probability as it is more likely that the sensors fail over a number of missions, at which point new sensors may be deployed in the subregion.

This assumption is also of value when the D-spectrum is utilized to estimate reliability, as it implies we do not have to repeatedly estimate the D-spectrum. Since the D-spectrum is independent of the failure distribution of sensors in the network, we are not required to constantly re-estimate the D-spectrum based on observing a different age composition of sensors. The only step that is required is to update the residual life distribution (30b), after which reliability can be estimated by applying (31). While estimating the D-spectrum is more efficient than a traditional Monte Carlo simulation to estimate reliability, repeatedly estimating the D-spectrum for different network structures becomes computationally burdensome. With the assumption that the network size informs the resulting network structure, we can estimate the D-spectrum for a wide range of network sizes (e.g., for a network with 300 to n^{max} sensors) once at the very beginning of the problem and store the estimates for use later in the ADP model.

Going forward, we adopt a network structure that is dependent upon a weight, w_i, assigned to each subregion. The purpose of assigning weights is to influence the distribution of sensors throughout the network, where increasing the weight assigned to a subregion corresponds to a larger proportion of the overall $\bar{N}_m^x$ sensors located within the subregion. For example, we may wish for a larger proportion of sensors to be located in subregions around the sink node compared to those farther away in an attempt to lower the possibility that a single sensor failure causes a large portion of the WSN to become disconnected. This behavior can be observed by increasing the weight on subregions surrounding the sink node. With weights properly assigned, even if a large number of sensors are observed in a given subregion there will still be new sensors deployed in the subregion to account for the possibility of sensor failures over the upcoming mission. In determining the network structure for a network of $\bar{N}_m^x$ sensors, the weight assigned to a subregion is inversely proportional to the distance from the sink node to the center of the subregion. Once the weights have been selected each subregion contains approximately $(w_i / \sum_{i=1}^{n_r} w_i)\bar{N}_m^x$ sensors. Given the observed number of sensors in a subregion N_m and decision to deploy $\bar{x}_m$ sensors, the deployment action x_m itself is now determined such that the resulting number of sensors in a subregion is approximately $(w_i / \sum_{i=1}^{n_r} w_i)\bar{N}_m^x$.

The primary decision is now how many new sensors to deploy in the network from which the resulting network size is determined, and informs which subregion the sensors will be deployed in. The number of sensors deployed can also be bounded between 0 and $\tilde{n}$. Based on the previous assumption that deploying a single additional sensor has a minor impact on network reliability and the future number of successful missions, we can also search the range $(0, \tilde{n})$ in an interval of d sensors to find the optimal action. Through the use of the D-spectrum and lookup tables we can now quickly evaluate the expression inside the maximization operator in (28), and select the action resulting in a maximum value.

4.3.4 Initializing the Value Function

A more simplistic policy considers the impact of deploying sensors on only the upcoming mission. This is a version of a myopic policy, and can be informative in our ADP formulation as well. Since a myopic policy is interested in reliability of a single mission, the policy will always deploy sensors until a constraint limits the action. That is, the myopic policy will never skip a deployment opportunity, and deploy sensors every mission until a constraint is reached (e.g., budget no longer available or maximum network size reached). When considering a myopic policy it is therefore more appropriate to consider, or allocate, a small budget to each mission to ensure there is a budget available to missions near the end of the planning horizon as well. A myopic CBDP is explored in [35], and is of value to our ADP model in two ways. First, as discussed in [35], a myopic CBDP allows greater focus on where sensors are deployed (e.g., which subregion) which contributes to the search for a network structure that maximizes reliability based on the number of sensors. While determining an optimal network structure remains a difficult task, a number of different policies anticipated to result in a highly reliable network can be compared efficiently to select the best policy. This methodology supports the decision to define subregion weights based on the distance from the sink node as described in Section 4.3.3 when characterizing the network structure for a given network size. Second, the resulting reliability estimate of a myopic CBDP can be of value in the ADP formulation to initialize the value function. In the ADP problem, if there is a budget B_m remaining then one option is to evenly allocate this budget to the remaining $M-1-m$ missions. This essentially corresponds to a myopic policy with each mission receiving $\frac{B_m}{M-1-m}$ of the budget. The reliability of the myopic policy can then be used to estimate the number of successful missions in the remaining $M-1-m$ missions and initialize the value function.

4.3.5 Approximate Value Iteration Algorithm

Algorithm 3 outlines an approximate value iteration (AVI) algorithm utilizing a value function approximation based on a lookup table representation on the aggregated state space, adapted from [28]. The AVI algorithm updates our value function approximation over a sequence of iterations $y = 1, 2, \ldots, Y$, which in turn updates the CBDP. S_m^y represents the observed state at the beginning of mission m in iteration y, and $S_m^{x,y}$ represents the post-decision state variable given action x_m. $\bar{V}_{m-1}^y(S_{m-1}^{x,y})$ represents the value function approximation for the post-decision state variable $S_{m-1}^{x,y}$ during iteration y, and is updated based on the step size parameter η_y. While Algorithm 3 outlines a relatively standard AVI algorithm, we hope to show that the resulting CBDP are a significant improvement over both a myopic condition-based deployment policy and a time-based deployment policy. As this is also one of the first ADP applications for the maintenance of a complex WSN with respect to a reliability evaluation, the performance of the AVI algorithm can also identify components of the model to focus more on in future work.

4.4 Numerical Example

In this section we illustrate the performance of the ADP formulation and provide results for a number of test instances. To model the failure of sensors, the lifetime of each sensor is distributed according to a Weibull distribution with a shape parameter $\beta = 1.5$ and scale parameter $\lambda = 10$. Sensor capabilities are defined by on a common communication radius $d_1 = 0.075$ and a monitoring radius of $d_2 = 0.075$. The cost of deploying sensors in the network is determined by the variable cost $c_V = 1$, with a fixed cost $c_F = 100$ incurred each

Algorithm 3 AVI for Finite Horizon Problem Using the Post-Decision State

1: **function** AVI
2: Initialization: approximation of the value function $\bar{V}_m^0(S_m^x)$ for all post-decision states, and an initial state $S_0^{x,1}$. Set $y = 1$.
3: For $m = 0, 1, 2, \ldots, M-1$,
4: Determine $\hat{v}_m^y$ by

$$\hat{v}_m^y = \max_{x_m \in \mathcal{X}_{S_m}} \left(R_m(S_m^y, x_m) + \bar{V}_m^{y-1}(S_m^{x,y})\right)$$

and let x_m^y be the optimal action.
5: Update $\bar{V}_{m-1}^{y-1}$ using

$$\bar{V}_{m-1}^y(S_{m-1}^{x,y}) = (1 - \eta_{y-1})\bar{V}_{m-1}^{y-1}(S_{m-1}^{x,y}) + \eta_{y-1}\hat{v}_m^y.$$

6: Sample W_{m+1}^y and compute the next state $S_{m+1}^y = S^M(S_m^y, x_m^y, W_{m+1}^y)$.
7: Increment y. If $y \leq Y$ go to step 3.
8: Return the value functions $(\bar{V}_m^n)_{m=0}^{M-1}$.
9: **end function**

time one or more sensors are deployed. The region of interest is a $[0,1] \times [0,1]$ unit square that is partitioned into $n_r = 16$ equal sized subregions of size 0.25×0.25. Additionally, 441 targets are uniformly spaced as a 21×21 grid representing target locations where the WSN must provide coverage.

The step size influences the rate at which the value function approximation is updated and the convergence of the AVI algorithm. Since the value functions are initialized with a myopic CBM policy, the initial step size for updating the value function approximation is $\eta_0 = 0.7$, and the step size is updated according to

$$\eta_y = \eta_0 \frac{a}{a + y - 1}, \tag{32}$$

with $a = 20$. This step size rule allows the rate at which η drops to zero to be influenced by the parameter a, with larger values slowing the rate at which η decreases.

For the test instances, the inspection interval δ varies among $\{2, 3, 4\}$, and the number of missions is selected so that the total time horizon $(M * \delta)$ is approximately the same. The coverage requirement is set at $\alpha = 0.8$, meaning if the WSN covers less than 80% of target locations the network is in a 'failed' state. The maximum network size is also fixed at $n^{max} = 950$ sensors for every test instance, with an initial number of $\bar{N}_0 = 650$ sensors deployed in the region. Parameter values for each test instance, to include the starting budget, B_0, and reliability requirement, ϕ, are provided in Table 3. To force exploration in the decision space, each mission there is a 5% chance a random non-optimal deployment action is implemented.

Table 3: Test Instances and Policy Performance

δ	M	B_0	ϕ	V_0	MC-PE
4	25	8700	0	24.97	24.95
4	25	8700	0.95	24.97	24.97
4	25	7600	0	23.99	23.85
4	25	7600	0.89	23.66	23.66
4	25	7400	0	23.13	22.69
4	25	7400	0.79	22.97	22.65
3	33	8050	0	31.89	31.71
3	33	8050	0.85	31.88	31.69
3	33	7650	0	29.45	28.14
3	33	7650	0.65	26.27	27.42
2	50	8700	0	49.95	49.89
2	50	8700	0.95	49.96	49.94
2	50	7600	0	48.54	47.54
2	50	7600	0.89	48.05	46.73
2	50	7400	0	47.19	45.55
2	50	7400	0.79	46.33	44.89

Table 3 also provides performance results of Algorithm 3 with $Y = 300$ replications, where column 5 (labeled V_0) reports the expected number of successful missions from the resulting

ADP policy. The final column in the table, labeled Monte Carlo Policy Evaluation (MC-PE), reports the average number of successful missions observed when the optimal ADP policy is evaluated through a Monte Carlo Simulation, assisting a later discussion on a comparison of the expected vs observed policy performance. Starting with $\delta = 4$ and the largest budget $B_0 = 8700$, the WSN is not overly strained and a sufficient number of new sensors can be deployed when needed to maintain the WSN at a high level. The budget is also large enough that enforcing a minimum reliability requirement on every mission has little impact on the performance of the optimal deployment policy. The next pair of test instance reduces the budget by $1,100$ which corresponds to a smaller number of sensors that can be deployed, and a larger emphasis on deploying sensors effectively to avoid the fixed cost consuming a large portion of the budget. While the budget is more constraining in this instance, the expected number of successful missions of 23.66 (23.99 without a reliability requirement) is still relatively high. The following pair of test instances result in a similar decline in WSN performance, particularly when a reliability requirement is present. Compared to the previous group of test instances the budget has decreased slightly to $7,400$, while the decline in the expected number of successful missions is comparable to lowering budget from $8,700$ to $7,600$. This pair of test instances also help illustrate the value in providing a minimum reliability requirement for each mission. When no requirement is imposed and there is no penalty for WSN failure then network reliability for a given mission can be sacrificed to avoid the fixed cost. This allows a larger number of sensors to be deployed over the remaining missions. When the reliability requirement is set to $\phi = 0.79$ this ensures that the probability a single mission is successful is still relatively high and also has little impact on the expected number of successful mission over the planning horizon.

In the next grouping of test instances the inspection interval is lowered to $\delta = 3$, and for the total time horizon to remain approximately the same the planning horizon for the number of missions is increased to 33. The noticeable result from this grouping is again observed in the smallest budget instance with a reliability requirement in place. With a budget of $7,650$ and a minimum reliability requirement of $\phi = 0.65$, the expected number of successful missions is significantly smaller compared to the case when no requirement is in place. This is again a result of not penalizing WSN failure, and by sacrificing performance to avoid incurring the fixed cost the budget for the remaining missions is large enough to maintain a highly reliable network.

The last grouping of test instances contain the shortest inspection interval with $\delta = 2$ and the largest number of missions with 50, influencing the policy in a number of areas. With a smaller inspection interval the network is observed more frequently, and there is an opportunity to observe a network state that might fail during the next mission that would not be observed under a larger inspection interval. In this scenario, new sensors can be deployed to avoid the potential network failure, and the overall number of successful missions should increase. Alternatively, with a shorter time between inspections it might be more advantageous to avoid deploying sensors in the network if the reliability of the upcoming mission is already at a sufficient level. While this does not improve reliability for the next mission, the fixed cost is avoided and allows a larger number of sensors to be deployed in the network over the remaining missions. For the largest starting budget of $8,700$ the ADP policy again results in an expected number of successful missions that is near the total number. Even though the smaller inspection interval results in more frequent network observation and more flexibility in when sensors are deployed, the decline in the expected number of successful missions as the starting budget decreases remains noticeable.

4.4.1 Monte Carlo Policy Performance

The optimal CBDP identified by the ADP algorithm is also implemented in a Monte Carlo simulation to observe the average number of successful missions the policy achieves, and is reported in the "MC-PE" (Monte Carlo Policy Evaluation) column of Table 3. These results help demonstrate the performance of the deployment policy in a simulated setting obtain results close to the predicted values. In several of the test instances with a larger inspection interval the performance of the ADP policy matches the expected number of successful missions. The largest difference between the expected and observed number of successful missions occurs for the smallest budget and smallest inspection interval test instance. In this test instance, the observed number of successful missions is slightly smaller than the expected number. Observing the largest deviation in this test instance is somewhat expected since this corresponds to a more difficult scenario. A smaller δ results in more missions, which implies a larger number of decisions are made. This instance is also more resource constrained since it has the smallest budget. While the observed performance of the ADP policy does begin to deviate as the test instances become more difficult, the overall observed number of successful missions remains relatively high.

The observed MC-PE also provides a more appropriate comparison on the results for an inspection interval of $\delta = 4$ with an inspection interval of $\delta = 2$. For each test instance, the resulting ADP policy with an inspection interval of $\delta = 4$ is a also a feasible policy for the corresponding $\delta = 2$ test instance. As a result, the observed number of successful missions in an optimal ADP policy for the $\delta = 2$ instance should be at least twice that of the corresponding $\delta = 4$ test instance. However in a majority of the test instances the observed number of successful missions for the $\delta = 2$ ADP policy is approximately double that of the corresponding $\delta = 4$ ADP policy, and is lower than expected in the $B_0 = 7400, \phi = 0.89$ test instance. This again highlights the difficulty of the test instance and the impact of reducing the time between network observations. When the network is inspected more frequently a larger number of deployment decisions must be made regarding when and how many sensors are deployed. The comparison in the observed performance of the ADP policy for different inspection intervals further demonstrates the complexity of a policy related to the repeated deployment of sensors in a WSN, and suggest there is an opportunity for future work to focus on improving a policy when the planning horizon increases.

4.4.2 ADP Comparison to Myopic Policy

In addition to initializing the value function, the myopic deployment policy provides a good comparison to demonstrate the improvement of the ADP policy. For this purpose, the myopic CBDP is also implemented in a Monte Carlo simulation with a budget of B_0/M available to deploy sensors per mission. The observed number of successful missions for the myopic policy is provided in Table 4, along with the previous ADP results.

Table 4: Observed ADP and Myopic Policy Comparison

δ	M	B_0	ϕ	ADP Policy	Myopic Policy
4	25	8700	0.95	24.97	23.96
4	25	7600	0.89	23.66	21.44
4	25	7400	0.79	22.65	19.34
3	33	8050	0.85	31.69	28.31
3	33	7650	0.65	27.42	18.98

In each of the test instances the ADP policy results in a larger number of successful missions, and is more noticeable with a smaller budget. This result is somewhat expected since the ADP policy is allowed to deploy a variable number of sensors and reallocate the budget as necessary, saving when able and deploying a larger number of sensors when needed. However the magnitude of this improvement is quite significant particularly when the budget is more constraining, clearly seen in the instance with $\delta = 3$ and a budget of $B_0 = 7650$. With the small budget available in this instance the myopic policy performs quite poorly and only 19 of the 33 missions are successful, compared to the ADP policy which is able to achieve over 27 successful missions. A similar outcome is observed with an inspection interval of $\delta = 4$, in which the ADP policy again performs noticeably better than the myopic policy in each instance. The significant improvement of the ADP policy over a myopic policy illustrates the value of a deployment policy that considers the impact on network performance over a planning horizon, compared to traditional policies that focus on an immediate effect.

4.4.3 ADP Policy Investigation

We are also interested in investigating the impact any test instance parameters have on the resulting ADP Policy. One observation is that the optimal policy is more likely to skip a deployment opportunity (i.e., deploy zero sensors at the start of a mission) as the starting budget B_0 and/or the inspection interval δ decrease. For a large starting budget, it may be possible to incur the fixed cost every mission and still deploy a sufficient number of sensors to maintain a highly reliable network. As the budget decreases, the fixed cost of deploying sensors every mission consumes a larger proportion of the overall budget which results in fewer sensors deployed each mission. Therefore, it becomes more desirable to skip a maintenance opportunity when allowed to avoid the fixed cost, providing a larger budget over the remaining missions and increasing the proportion of the budget consumed by the variable cost, which equates to a new sensor in the network. Similarly, as the inspection interval decreases the amount of time the network must function until the next deployment window is also smaller. Compared to a larger inspection interval, it is likely that fewer sensors will fail in a shorter time interval and the network will more often be observed in a state providing the opportunity to skip sensor deployment while ensuring the upcoming mission remains successful with high probability.

Table 5: Percent of Budget Dedicated to Variable Cost

δ	M	B_0	No Reliability Requirement	With Reliability Requirement
4	25	8700	71.26%	71.41%
4	25	7600	68.42%	67.25%
4	25	7400	68.52%	67.38%
3	33	8050	61.00%	61.02%
3	33	7650	62.51%	61.24%
2	50	8700	70.75%	70.61%
2	50	7600	69.53%	69.71%
2	50	7400	69.94%	70.21%

The average percent of the budget consumed by the variable cost in each policy is reported in Table 5. For each test instance the column labeled "No Reliability Requirement" implies $\phi = 0$, while the column "With Reliability Requirement" refers to the non-zero reliability requirement for the corresponding test instance defined in Table 3. When $\delta = 4$, the significant drop in the starting budget between the first and second test instance impacts both the total number of missions in which sensors are deployed and the number of sensors deployed. However given the longer time between inspection intervals it is more difficult to skip a deployment opportunity and maintain a highly reliable network, which is observed by the decrease from 71.26% to 68.42% (71.41% to 67.25% with a reliability requirement) of the the overall budget dedicated to variable cost. Meanwhile, the budget allocation appears to be impacted less for the smaller inspection intervals. For example, when $\delta = 3$ the overall proportion of the budget consumed by the variable cost is approximately the same when the starting budget decreases from $8,050$ down to $7,650$. Additionally, for the inspection interval $\delta = 2$ the decrease in the percent of budget allocated to the variable cost is not as significant compared to the larger interval of $\delta = 4$. This result is somewhat expected since the network does not have to operate as long until the next deployment decision, and there is more flexibility for the ADP policy to control when sensors are deployed in the network providing a better balance between the fixed and variable cost.

The discussion at the end of Section 4.4.1 also highlighted the difficulty encountered in the $\delta = 2, B_0 = 7400, \phi = 0.79$ test instance. Compared to the corresponding test instance with $\delta = 4$, a larger proportion of the overall budget is allocated to the variable cost under the smaller inspection interval of $\delta = 2$. This suggests that, as expected, the ADP policy in the $\delta = 2$ instance is skipping a deployment opportunity more often, but based on the observed policy performance compared to the $\delta = 4$ policy is struggling to do so in the most effective manner. This suggests that the ADP policy can potentially be improved by focusing more on the timing of when a deployment opportunity is skipped.

It is also interesting to note that for the smaller starting budgets and $\delta = 3$ or $\delta = 4$, the variable cost consumes a larger proportion of the budget when there is no reliability requirement present. The reason for this is that the ADP policy is actually more likely to skip a deployment opportunity when there is no minimum reliability to maintain. With no penalty for network coverage falling below the requirement and no minimum reliability the network must maintain the ADP policy is freely able to sacrifice network performance. By avoiding the deployment costs for the current mission, there is a larger budget for the remaining missions which likely contributes to an increase in the number of sensors deployed. When there is a minimum reliability requirement the policy must be more strategic in when a deployment opportunity is skipped to ensure reliability of every mission is sufficiently high. As a result, the opportunity to skip a deployment window likely arises by deploying a larger number of sensors at the beginning of a previous mission, and/or a favorable network observation in which only a small number of sensors failed during the prior mission. Compared to an instance with no reliability requirement, where an increase in the overall number of successful missions can be achieved by low network performance over one or more missions.

4.4.4 Single Region Comparison

Finally, we explore the influence specifying the subregion a sensor is deployed in has on the overall number of successful missions. A simpler strategy to implement might involve randomly deploying a sensor over the entire region of interest, and is one of the more common assumptions when deploying a WSN [6, 36]. The previous model formulation can easily address a single region by setting $n_r = 1$. It is interesting to note that since we previously defined a network structure by assigning weights to every subregion which determined how new sensors were deployed, a decision in the multiple subregion model is no more complex than the single subregion case. The only difference is that now sensors are randomly deployed over the entire region, whereas we previously used a rule-set to determine how sensors were allocated to each subregion.

Table 6 contains the expected number of successful missions from the optimal ADP policy when sensors are randomly deployed over the entire region. The final two columns of Table 6, under the 'Subregion' label, contain the results from the corresponding test instance with multiple subregions originally reported in Table 3. As expected, removing the ability to specify the subregion a sensor is deployed in lowers the expected number of successful missions compared to the original performance with multiple subregions. Even if the state variable definition remains the same (i.e., we are still able to observe the number and ages of sensors in various subregions in the network), there is now no guarantee that deploying new sensors based on observing a small number of sensors in one or more subregions at the beginning of a mission will improve the performance in the degraded areas of the WSN.

Table 6: Single Region Policy Performance

				Single Region		Subregion	
δ	M	B_0	ϕ	V_0	MC-PE	V_0	MC-PE
4	25	8700	0.95	24.91	24.89	24.97	24.97
4	25	7600	0.89	22.59	22.40	23.66	23.66
4	25	7400	0.79	20.79	21.12	22.97	22.65
3	33	8050	0.85	30.55	30.52	31.88	31.69
3	33	7650	0.65	24.53	25.35	26.27	27.42
2	50	8700	0.95	49.88	49.84	49.96	49.94
2	50	7600	0.89	45.73	44.03	48.05	46.73
2	50	7400	0.79	42.67	43.39	46.33	44.89

The decrease in expected number of successful missions resulting from randomly deploying sensors over the entire region compared to a smaller defined subregion is more noticeable for the smaller starting budgets. This can partially be attributed to the impact influencing network topology has on the probability of mission success in a smaller sized network compared to the impact in a larger network. In terms of the budget available, a decrease to the budget results in a decrease in the total number of sensors that are deployed over the planning horizon, and as a result the overall size of the WSN is generally smaller as well. For smaller sized networks it is less likely that randomly deploying sensors over the entire region of interest will result in sensors sufficiently distributed throughout the region for coverage purposes, and within the communication radius of nearby sensors necessary to route information to the sink node. While randomly deploying a sensor within a smaller subregion does not entirely remove this problem, it does provide the ability to avoid the situation in which one portion of the WSN is overly dense with sensor nodes whereas another portion of the network is uncovered and individual sensors are isolated. Therefore, there is a larger benefit (e.g., improvement in probability of mission success) in a smaller network when the subregion a sensor is deployed in can be specified compared to the benefit present in a larger sized network. This is observed several of the test instances, for example with $\delta = 4$ and $B_0 = 7400$ where the single region ADP policy achieves an expected number of successful missions of 20.79, while the previous results with 16 subregions achieve an optimal ADP policy with an expected 22.97 successful missions. Additionally, even if there is only a minor improvement for a single mission the cumulative impact over the entire planning horizon can be more substantial.

Exploring the performance in a single region model helps further illustrate the significance of the ADP policy and considering the impact of an action on future missions as well. Notice that the observed performance of the single region ADP policy, reported in the 'MC-PE' column of Table 6, is still able to outperform the myopic condition-based policy. This highlights the advantage of deciding if and how many sensors are deployed each mission, allowing an appropriate allocation of the budget to each mission as necessary. Even if new sensors are randomly deployed over the entire region of interest, rather than more controlled through a subregion deployment policy, the decision on when and how many sensors are deployed has a significant impact on WSN performance over an extended period of time.

A single region scenario also enables a more straightforward comparison with the TBDPs considered in [34], where sensors are deployed in order to restore the network to a fixed network size at periodic time intervals. Instead of a direct comparison with a TBDP, we can first note that there exists a close relationship between a TBDP and a corresponding myopic CBDP. In [34] an expression for the cost rate of an associated TBDP is derived based on the expected number of sensors that fail during a mission. The expected number of failed sensors informs the average cost of deploying sensors to reach a fixed network size, which can now be treated as a fixed budget available in a myopic CBDP. A TBDP differ from the myopic CBDPs in Section 4.4.2 in that sensors are randomly deployed over the entire region rather than a specified subregion. Since the myopic CBDP provides more control over how sensors are deployed, the performance of a myopic CBDP is at least as good as the related TBDP. With this similarity, and the previous discussion on the improvement of a single region ADP policy over a myopic CBDP, the ADP policy also improves upon a simpler time-based policy.

4.5 Conclusion

The coverage and communication capability of a WSN is made possible through the cooperative effort of a large number of sensor nodes. The flexibility with which WSNs can be established, randomly deploying sensors over a target region when exact placement is not feasible, enables their incorporation into a wide range of applications. It is important to consider not only the initial capability provided by a WSN, but performance over a period of time and the impact of eventual sensor failures. As the number of failed sensors increases the decline in network capability becomes more significant and appropriate actions must be taken to restore WSN coverage and communication abilities. A large focus on research related to this problem has been on deploying a small number of new sensor nodes in the network at a single point in time. The maintenance of a WSN over a prolonged period time in which sensors are repeatedly deployed in the network has received less attention.

In this work we have contributed an MDP model for the condition-based sensor deployment problem in which new sensors are deployed in the network over an extended period of time. While MDP models have been applied to a wide range of WSN related problems, our model is one of the few addressing maintenance through the repeated deployment of new sensor nodes, and one of the first ADP applications for the maintenance of a complex WSN. Whereas previous sensor deployment models have primarily been interested in extending a network lifetime metric, our work also addresses the complexity encountered by incorporating a reliability objective. A few of the difficulties that must be addressed in this problem include a variation in the age composition of sensors as well as a dynamic network topology as sensors fail and new sensors are deployed in the network. Our methodology has addressed both of these issues by the incorporation of the network D-spectrum. The D-spectrum has been widely research in network reliability problems, but only a handful of works discuss the D-spectrum in a maintenance optimization model as well [33–35]. Finally, we discussed an ADP solution approach using a value function approximations to determine optimal CBDPs, and presented results on a range of test instances.

The model also provides several directions for future work, focusing both on the modeling assumptions and ADP methodology discussed in Section 4.2. The reliability of a WSN is currently defined based on a given coverage requirement. The objective is to maximize reliability, but there is otherwise no detriment to not satisfying the coverage requirement over a mission. One possibility is to include a penalty based on the probability of network failure, which could also reflect need for immediate maintenance to provide a functioning WSN at all times.

The current model also assumes the WSN is observed every δ time units and does not explicitly incorporate any cost associated with observation. A more complex decision might include whether the WSN is inspected/observed or not, where there is a cost associated with observing the network. Similarly network observation may be imperfect or there might be a time delay between our observation and deployment action. These directions begin to incorporate uncertainty in the true state of the network at the time sensors are deployed and might be better modeled as a partially observable MDP.

Our value function approximation was based on a combination of aggregation functions and lookup tables. Future work might consider the use of several basis functions and building a parametric model to approximate the value function. In this approach the previously defined aggregation functions may still be of use, but exploration is needed to define additional basis functions and an appropriate model representation (e.g., linear, nonlinear, etc.). A parametric model approximation of the value function is also of interest because it may provide additional opportunities to solve the optimality equation each stage, allowing the optimal action to determined more efficiently.

References

[1] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 1, pp. 5:1–5:39, Feb. 2009.

[2] T. Arampatzis, J. Lygeros, and S. Manesis, "A survey of applications of wireless sensors and wireless sensor networks," in *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.* IEEE, 2005, pp. 719–724.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, 2002.

[4] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[5] S. Debnath, A. K. Singh, and A. Hossain, "A comprehensive survey of coverage problem and efficient sensor deployment strategies in wireless sensor networks," *Indian Journal of Science and Technology*, vol. 9, no. 45, pp. 1–6, 2016.

[6] M. Ishizuka and M. Aida, "Performance study of node placement in sensor networks," in *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.* IEEE, 2004, pp. 598–603.

[7] L. M. S. De Souza, H. Vogt, and M. Beigl, "A survey on fault tolerance in wireless sensor networks," *Interner Bericht. Fakultät für Informatik, Universität Karlsruhe*, 2007.

[8] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 1, pp. 89–124, 2005.

[9] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems.* New York, NY: ACM, 2003, pp. 28–39.

[10] L. Frye, L. Cheng, S. Du, and M. W. Bigrigg, "Topology maintenance of wireless sensor networks in node failure-prone environments," in *2006 IEEE International Conference on Networking, Sensing and Control.* IEEE, 2006, pp. 886–891.

[11] N. Li and J. C. Hou, "Flss: a fault-tolerant topology control algorithm for wireless networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking.* ACM, 2004, pp. 275–286.

[12] A. Sen, B. H. Shen, L. Zhou, and B. Hao, "Fault-tolerance in sensor networks: A new evaluation metric," in *INFOCOM 2006: 25th IEEE International Conference on Computer Communications*, 2006, p. 4146923.

[13] S. Parikh, V. M. Vokkarane, L. Xing, and D. Kasilingam, "Node-replacement policies to maintain threshold-coverage in wireless sensor networks," in *2007 16th International Conference on Computer Communications and Networks.* IEEE, 2007, pp. 760–765.

[14] A. Jain and B. Reddy, "Node centrality in wireless sensor networks: Importance, applications and advances," in *2013 3rd IEEE International Advance Computing Conference*. IEEE, 2013, pp. 127–131.

[15] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Networks*, vol. 14, no. 3, pp. 347–355, 2008.

[16] E. L. Lloyd and G. Xue, "Relay node placement in wireless sensor networks," *IEEE Transactions on Computers*, vol. 56, no. 1, pp. 134–138, 2006.

[17] J. L. Bredin, E. D. Demaine, M. Hajiaghayi, and D. Rus, "Deploying sensor networks with guaranteed capacity and fault tolerance," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2005, pp. 309–319.

[18] H. M. Almasaeid and A. E. Kamal, "On the minimum k-connectivity repair in wireless sensor networks," in *2009 IEEE International Conference on Communications*. IEEE, 2009, pp. 1–5.

[19] C. R. Cassady, E. A. Pohl, and W. P. Murdock, "Selective maintenance modeling for industrial systems," *Journal of Quality in Maintenance Engineering*, vol. 7, no. 2, pp. 104–117, 2001.

[20] C. R. Cassady, W. P. Murdock Jr, and E. A. Pohl, "Selective maintenance for support equipment involving multiple maintenance actions," *European Journal of Operational Research*, vol. 129, no. 2, pp. 252–258, 2001.

[21] L. M. Maillart, C. R. Cassady, C. Rainwater, and K. Schneider, "Selective maintenance decision-making over extended planning horizons," *IEEE Transactions on Reliability*, vol. 58, no. 3, pp. 462–469, 2009.

[22] K. Ahadi and K. M. Sullivan, "Approximate dynamic programming for selective maintenance in series–parallel systems," *IEEE Transactions on Reliability*, vol. 69, no. 3, pp. 1147–1164, 2019.

[23] M. A. Alsheikh, D. T. Hoang, D. Niyato, H.-P. Tan, and S. Lin, "Markov decision processes with applications in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1239–1267, 2015.

[24] S. Misra, S. R. Mohan, and R. Choudhuri, "A probabilistic approach to minimize the conjunctive costs of node replacement and performance loss in the management of wireless sensor networks," *IEEE Transactions on Network and Service Management*, vol. 7, no. 2, pp. 107–117, 2010.

[25] J. S. Provan and M. O. Ball, "The complexity of counting cuts and of computing the probability that a graph is connected," *SIAM Journal on Computing*, vol. 12, no. 4, pp. 777–788, 11 1983.

[26] W. B. Powell, "What you should know about approximate dynamic programming," *Naval Research Logistics*, vol. 56, no. 3, pp. 239–249, 2009.

[27] ——, "Perspectives of approximate dynamic programming," *Annals of Operations Research*, vol. 241, no. 1, pp. 319–356, 2016.

[28] ——, *Approximate Dynamic Programming: Solving the curses of dimensionality.* John Wiley & Sons, 2007, vol. 703.

[29] Y. Shpungin, "Networks with unreliable nodes and edges: Monte carlo lifetime estimation," *Applied Mathematics and Computer Science*, vol. 27, pp. 168–173, 2007.

[30] M. O. Ball, C. J. Colbourn, and J. S. Provan, "Network reliability," *Handbooks in Operations Research and Management Science*, vol. 7, pp. 673–762, 1995.

[31] D. Shier and N. Liu, "Bounding the reliability of networks," *Journal of the Operational Research Society*, vol. 43, no. 5, pp. 539–548, 1992.

[32] F. J. Samaniego, "On closure of the ifr class under formation of coherent systems," *IEEE Transactions on Reliability*, vol. R-34, no. 1, pp. 69–72, April 1985.

[33] M. Finkelstein and I. Gertsbakh, "Time-free preventive maintenance of systems with structures described by signatures," *Applied Stochastic Models in Business and Industry*, vol. 31, no. 6, pp. 836–845, 2015.

[34] N. T. Boardman and K. M. Sullivan, "Time-based node deployment policies for reliable wireless sensor networks," *IEEE Transactions on Reliability*, pp. 1–14, 2021.

[35] ——, "Condition-based node deployment policies for reliable wireless sensor networks," University of Arkansas, Tech. Rep., 2021.

[36] M. R. Senouci, A. Mellouk, and A. Aissani, "Random deployment of wireless sensor networks: a survey and approach," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 15, no. 1-3, pp. 133–146, 2014.

[37] J. Navarro, F. J. Samaniego, N. Balakrishnan, and D. Bhattacharya, "On the application and extension of system signatures in engineering reliability," *Naval Research Logistics*, vol. 55, no. 4, pp. 313–327, 2008.

[38] I. B. Gertsbakh and Y. Shpungin, *Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo.* Boca Raton, FL: CRC press, 2016.

[39] J. Navarro, F. J. Samaniego, and N. Balakrishnan, "Signature-based representations for the reliability of systems with heterogeneous components," *Journal of Applied Probability*, vol. 48, no. 3, pp. 856–867, 2011.

5 Maximizing the Expected Coverage of a Wireless Sensor Network under Stochastic Sensor Deployment

Abstract

The coverage achieved by a wireless sensor network (WSN) is often an important factor when deploying sensors over a region of interest. In certain applications network coverage can be influenced by placing sensors at specific locations throughout the region. Other environments may require sensors to be randomly deployed over the region, for example when deploying a WSN over an extremely large region or when the terrain is difficult to access. In this work we address the problem of a WSN randomly deployed over a region of interest in which the exact placement of a sensor is not possible but a smaller subregion of the network can be selected for the sensor to be deployed in. The main contribution is a stochastic optimization model to maximize the expected coverage of a WSN with uncertainty in the placement of each sensor. A scenario based approach is applied to randomly sample the location for each sensor and solve the model in order to determine an optimal deployment policy. To address a large number of scenarios we present a heuristic solution method, and compare the performance of our heuristic solution to both a random distribution of sensors over the entire region and a policy that distributes an equal number of sensors to each subregion. We also discuss an extension of the model to incorporate sensor failures and the impact on an optimal deployment policy. As the number of sensors deployed decreases or sensors in the network are subject to failure, the heuristic solution to the stochastic optimization model demonstrates a clear improvement over both the random and equal distribution policies.

5.1 Introduction

Wireless sensor networks (WSNs) can be broadly characterized by a number of sensor nodes deployed over a region of interest with a responsibility to monitor and report desired measures or events occurring within the region. The capability of an individual sensor is limited in that it can only monitor an area immediately surrounding the sensor and communicate directly with those in a small neighborhood. Transmitting information over a longer distance is made possible through a communication path that routes a message using a series of sensor nodes, often directed toward toward a sink node [1]. To monitor a large area this also requires that sensor nodes are distributed throughout the region while maintaining a degree of network connectivity. Depending on the environment the WSN is operating in this may be possible by locating sensors at specific points within the region. For example, a WSN established in a building for the purpose of regulating temperature in a heating/cooling system can place sensors at specific points to satisfy desired performance criteria [2].

When WSN topology can be controlled through the deployment of sensors at specific locations, an additional emphasis is commonly placed on determining the locations to deploy the fewest number of sensors while ensuring the network satisfies a coverage and/or connectivity requirement [3, 4]. One of the advantages of this objective is it helps limit deployment cost, but deploying the minimum number of sensors can be problematic for future WSN performance. While the network will initially satisfy a coverage/connectivity requirement, the performance of the WSN declines over time. This is attributed to the failure of sensors resulting from a finite power supply that must be consumed over the course of monitoring the surrounding area and communicating with nearby sensors [5]. One method to prolong the decline in WSN performance is to place redundant sensors in the network and achieve a level of k-connectivity [3]. When the network is easily accessible an alternative option is to deploy additional sensors in the network over time as necessary. This is the focus in [6] and [7], where the problem is to determine the fewest number of sensors and their locations in the network to restore network connectivity. Similar to the initial deployment of sensors, this problem can be expanded to deploy sensors based on maintaining a level of k-connectivity, explored in [8–10].

A large focus in sensor deployment related problems has been on deploying the fewest number of sensors to fully cover the region of interest or to result in a fully connected network. A similar problem is to deploy a fixed number of sensors to maximize coverage or connectivity. If coverage is the only concern and all sensors have an identical coverage radius, determining optimal sensor locations relates closely to the circle packing problem discussed in [11]. The problem addressing multiple types of sensors with different coverage radii was first explored in [12]. Due to the difficultly present in solving the problem exactly with different sensor types, the authors propose a genetic algorithm that determines the location for each sensor to maximize coverage. Heuristic algorithms for determining optimal locations with heterogeneous sensors have been explored further in [13–15], with efforts directed toward improving solution quality and reducing the computation time required. One of the limitations of the coverage model is it does not account for a communication radius for sensors or address the potential for failures which may result in a large drop in network coverage.

In other environments it may be impractical or costly to specifically locate sensors in the network. An attractive feature of WSNs is that a network can still be established in this situation by randomly deploying sensors over a target region. For example, when a WSN is located on a steep mountainside or over a dense forest, sensors can be dropped from a helicopter to quickly set up network capability [16]. This may require a larger number of sensors to be deployed to account for the randomness in where a sensor lands, but is made feasible through the low cost of an individual sensor [2], as well as the lack of infrastructure sensors require to communicate with each other [17]. If the sensors contain a mobile capability it may also be possible to reposition sensors after they are randomly deployed to improve coverage, addressed in [18]. However the ability of sensors to move greatly increases the cost of an individual sensor which deviates from the attractive low-cost characteristic of WSNs [19].

While managing network topology in a randomly deployed setting is more difficult, WSN coverage and connectivity measures can be influenced by controlling sensor density [20]. It is common to assume this density is constant throughout the network resulting in a random uniform distribution of sensors over the network, which we refer to as a uniform policy. An attractive feature of a uniform policy is that the probability of full network coverage can be estimated analytically [21]. In [22] a uniform policy is compared via simulation to a simple diffusion strategy, in which sensors are randomly scattered around a central location (for example the sink node), and a new strategy called R-random placement that emphasizes a larger density of sensors surrounding the sink node compared to locations farther away. A survey of random deployment strategies is provided in [23], in additional to a discussion on simulated results. In [24], random deployment is addressed by deploying sensors in stages. In the first stage a number of sensors are deployed according to a uniform policy and the locations are observed, which enables an initial estimate on WSN coverage and connectivity.

If the estimated performance is not satisfactory the process repeats and additional sensors are randomly deployed until performance reaches a desired level or the maximum number of sensors is reached.

Whether sensors are deployed in a deterministic or random manner, the resulting network coverage is a common performance measure of interest. Under both deterministic and random deployment network coverage is frequently used to approximate the network size necessary to fully cover the region of interest. While full region coverage is desirable, there are many applications in which partial coverage provides sufficient information to construct a picture of the overall region. For example, monitoring the temperature or humidity at every location in a region is often unnecessary, and measurements at a subset of the locations scattered over the region provides the desired information [2]. Alternatively, there may only be a limited number of sensors available (e.g., due to cost or inventory restrictions) and sensors must be deployed to maximize coverage of the resulting WSN [12]. Therefore, it might not be as important for the network to provide full coverage but rather that a large portion of the region is covered, and as long as some minimum coverage level is satisfied the network can still operate at a high level.

The previous discussion motivates the problem addressed in this work of deploying a fixed number of sensors in various subregions throughout the network with the objective of maximizing WSN coverage. Sensor deployment is influenced by selecting a subregion of the the network a sensor is deployed in, but the exact location a sensor lands in a subregion is random. Therefore, while the WSN can be influenced by varying the number of sensors in different subregions the resulting WSN topology is uncertain. We formulate this as a stochastic optimization model that maximizes expected coverage with uncertainty in the final location of each sensor. To determine an optimal deployment policy (i.e., how many sensors are deployed in each subregion), a scenario based approach is presented which samples a location for each sensor randomly deployed in a subregion, allowing the resulting WSN topology to be constructed. In order to solve the model when a large number of scenarios are generated we present a heuristic approach to solve for an optimal deployment policy and discuss the heuristic performance on a wide range of test instances. The expected coverage of our heuristic solution is compared to a uniform policy, which is the most common model for randomly deployed sensors. Lastly, we discuss a simple extension of the optimization model to address the possibility of sensor failures as well.

Our work relates to [22] and [23] in that we focus on the random deployment of sensor nodes and allow for a more controlled deployment of sensors compared to a simple uniform policy. Similar to [12–15] we seek to deploy a fixed number of sensors in the network, but differ in that we account for a communication radius limiting the ability of sensors to route information through the network. The main contribution of this work is a stochastic optimization model to address the optimal deployment of sensors when the exact location of each sensor cannot be selected. While the optimal deployment of sensors in a deterministic setting has received significant attention, there appears to be far less in the optimal deployment of sensors to maximize coverage under a random deployment. The focus in [22] and [23] is primarily on evaluating and comparing different policies, and while they each discuss a new policy that improves desired performance metrics they do not further explore an optimal policy nor discuss optimal parameter settings related to the respective policy. Nonuniform random policies are also investigated in both [25] and [26], however the goal is to balance the rate of energy consumption for every sensor node. That is, the objective is to deploy a larger number of sensors near the sink node to share the responsibility of routing messages but the policy is designed so that every sensor fails at approximately the same time. A policy that balances energy consumption or maximizes the time to first sensor failure might be of value if full region coverage is required, but our model is concerned with maximizing coverage particularly when full coverage is not attainable. It may also be possible to deploy additional sensors in the network to restore communication abilities with isolated sensor nodes. Meanwhile, an optimization model is addressed in [27], however it is assumed that all sensors are able to communicate directly with a central sink node, and a single target is randomly located in the region. Lastly, our formulation assumes that all sensors are deployed at the same time, compared to the sequential approach in [24] which allows a network to be constructed over a period of time based on a previous observation.

The remainder of this work is organized as follows. Section 5.2 outlines the problem formulation, modeling of sensor location uncertainty and scenario construction, followed by our stochastic optimization model. Section 5.3 briefly discusses an exact solution methodology and outlines a heuristic procedure to determine a deployment policy maximizing expected coverage. Section 5.4 presents results on the expected coverage achieved for a wide range of network sizes, both with and without sensor failures, and explores characteristic of the deployment policy. The heuristic solution is also compared against both a uniform policy, and a policy that evenly distributes sensors throughout the network. Finally, Section 5.5 concludes this article.

5.2 Problem Formulation

In this section we present our WSN construction and modeling assumptions. We consider a WSN $\mathcal{G}$ to be deployed over a region of interest $\mathcal{R}$, partitioned into a number of smaller subregions represented by the indexed set $\mathcal{R}_s = \{1, 2, \ldots, R\}$. Given a number of sensors n to deploy in the WSN, the problem is to determine how many sensors are deployed in each subregion to maximize network coverage. It is assumed that the subregion a sensor is deployed in can be selected, but the sensor is then randomly deployed in the subregion. While the number of sensors deployed in each subregion can vary greatly, a maximum of $N < n$ sensors can be deployed in a subregion. N can be set large enough to not limit the deployment of sensors in an optimal solution, but is used later on to determine the maximum number of locations that need to be sampled in a subregion. In addition to the n sensor nodes, a single sink node is deployed in $\mathcal{R}$ as well.

A sensor is capable of communicating with any other sensor in the network, to include those deployed in different subregions and the sink node, provided the distance between the two nodes is less than or equal to a communication radius d_1. Sensors are additionally capable of monitoring any target within a monitoring radius d_2 of the sensor. For a target to be covered it must satisfy two criteria. First, there must be a sensor within the monitoring distance d_2 of the target. Second, there must exist a communication path from the monitoring sensor back to the sink node, either directly or through a communication path of several sensor nodes. Given a collection of targets in $\mathcal{R}$, network coverage is defined as the proportion of targets that are covered.

5.2.1 Maximum Flow Problem

In order to determine an optimal deployment of n sensors to maximize coverage we formulate a maximum flow problem where the value of the maximum flow represents WSN coverage. The uncertainty in sensor location is addressed by a set of scenarios, where a scenario consists of an observation on the random location for each sensor deployed in a subregion. Since a maximum of N sensors can be deployed in each subregion, this requires a sample of N locations in each of the R subregions to model the possible location of a sensor deployed in the subregion. Viewed in a slightly different manner, for each subregion a scenario consists of a location for the first sensor deployed in the subregion, the second sensor deployed in the subregion, continuing up to the Nth sensor deployed in the subregion. Given a decision n_r on the number of sensors deployed in subregion $r \in \mathcal{R}_s$, a scenario models the random deployment of n_r sensors in a subregion by selecting the first n_r locations sampled. With a scenario consisting of RN sampled locations, the problem is now re-framed in the context of determining which n sensors are deployed from the set of potential sensors $\mathcal{N}_1 = \{1, 2, \ldots, RN\}$, to maximize expected WSN coverage. Without loss of generality, we assume that the first group of N sensors correspond to sensors in subregion 1, the second group of N sensors correspond to sensors in subregion 2, and so on. The set of sensors in subregion r is therefore given by $\{(r-1)N+1, (r-1)N+2, \ldots, rN\}$, with the sink node represented by $\{0\}$. Provided a collection of m targets in the region, the set of target nodes is defined by the set $\mathcal{N}_2 = \{RN+1, RN+2, \ldots, RN+m\}$.

For every scenario ω, the WSN maximum flow network is now constructed as follows. First, for each sensor $i \in \mathcal{N}_1$ a location (x_i^ω, y_i^ω) is randomly sampled from the appropriate subregion representing where the sensor lands in the subregion. Based on the previous sensor grouping, this implies sensors 1 through N are randomly located in subregion 1, sensors $N+1$ through $2N$ are randomly located in subregion 2, etc. With a location for every sensor, the distance between two nodes (sensor, sink, and/or target) i and j, denoted $d^\omega(i,j)$, can be calculated. The sensor-to-sensor edge set reflects the ability of any two sensors deployed in the network within the appropriate range to communicate with one another, defined by the edge set $\mathcal{A}_1^\omega = \{(i,j) : d^\omega(i,j) \leq d_1, i \in \mathcal{N}_1, j \in \mathcal{N}_1, j \neq i\}$. The sensor-to-sink edge set, denoted $\mathcal{A}_2^\omega$, is constructed similarly, where $\mathcal{A}_2^\omega = \{(i,0) : d^\omega(i,0) \leq d_1, i \in \mathcal{N}_1\}$. The target-to-sensor edge set models the ability of a sensor to monitor any target within the monitoring radius and is defined by $\mathcal{A}_3^\omega = \{(i,j) : d^\omega(i,j) \leq d_2, i \in \mathcal{N}_2, j \in \mathcal{N}_1\}$. Lastly, an artificial source node $\{0'\}$ is introduced in the network, and a directed edge from the source node to each each target node is added defined by the edge set $\mathcal{A}_4 = \{(0',j) : j \in \mathcal{N}_2\}$. The set of all directed edges in the network for a scenario is denoted $\mathcal{A}^\omega = \{\mathcal{A}_1^\omega \cup \mathcal{A}_2^\omega \cup \mathcal{A}_3^\omega \cup \mathcal{A}_4\}$.

The decision is how many sensors are deployed in each subregion, represented by $n_r, r \in \mathcal{R}_s$, such that the total flow through the network averaged over every scenario is maximized. In addition to the number of sensors deployed in each subregion, there is a binary indicator variable $\gamma(i), i \in \mathcal{N}_1$, to capture if sensor i is deployed or not. This variable is used in a deployment constraint that forces the random location of n_r sensors deployed in a subregion to the first n_r sampled locations in the subregion for every scenario. That is, for every scenario the deployment constraint forces the first sensor deployed in the subregion to be at the first location sampled, the second sensor deployed at the second location sampled, and so on. This is equivalent to a constraint that only allows sensor i to be deployed in the network if sensor $i-1$ is also deployed. Similarly, the indicator variables help enforce that if sensor i

is deployed in one scenario then it is deployed in every scenario, and ensures consistency in the deployment decision across scenarios. The indicator variable $\gamma(i)$ is also used to enforce a constraint that the flow on any edge originating at a sensor node can only be positive (i.e., a sensor can only contribute to communication and monitoring responsibilities) if sensor i is deployed in the network.

An example of a network constructed for two different scenarios, excluding the source node and associated edges, is provided in Figure 10. The small examples are not necessarily drawn to scale and used primarily for illustration purposes. In this example there are four subregions (with borders indicated by a dashed line), and each subregion can contain a maximum of five sensors. Based on the sensor node labeling scheme, nodes $1-5$ corresponds to sensors located in subregion 1, nodes $6-10$ correspond to sensors located in subregion 2, $11-15$ sensors loacted in subregion 3, and $16-20$ sensors located in subregion 4. The purpose of the deployment constraint can be demonstrated in subregion 2. Notice that in scenario 1, target 22 can be covered by deploying three sensors to subregion 2. In this scenario the first three sensors deployed, sensors $6, 7,$ and 8, connect the target to the sink node and the target is covered. In scenario 2 however, the location of first three sensors are not as favorable and the target remains uncovered. Further, the deployment of a fourth sensors (sensor 9) still results in a lack of target coverage in scenario 2. Based on the scenarios generated, in order for target 22 to be covered in both scenario 1 and 2 the maximum limit of five sensors must be deployed in the second subregion.

Without the deployment constraint we may overestimate network coverage by selecting the more desirable sensor locations from those sampled. For example, in both scenario 1 and 2 the fourth location sampled (i.e., sensor 9), is in an unfavorable location and does

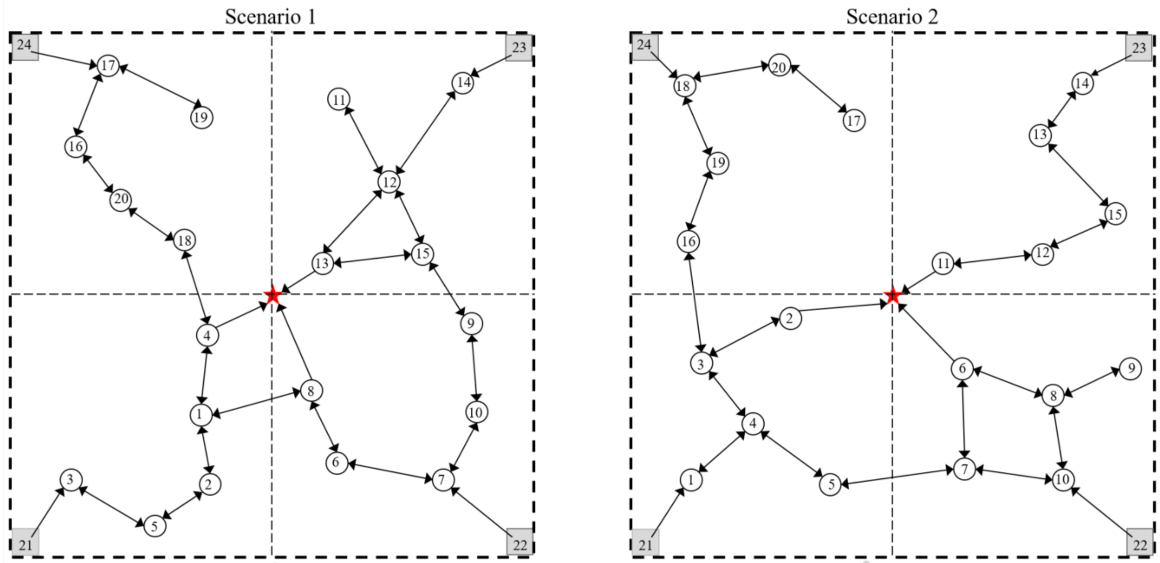

Figure 10: Example scenarios with sink node (⋆), sampled locations (○), and targets (■).

not contribute to network coverage. Note that this is less likely to occur as the number of scenarios increases since at least one scenario will likely sample a location for sensor 9 closer to the interior of the overall region, but is useful for discussion purposes. In the absence of the deployment constraint the model can 'skip' this sensor and deploy sensor 10 instead, resulting in the need to deploy only 4 sensors $(6, 7, 8,$ and $10)$ in subregion 2 to cover target 22 in both scenarios. The purpose of this placement constraint can therefore be viewed as capturing the randomness of deploying an additional sensor in a subregion, potentially landing in an area that achieves no increase in overall network capability, and prevents the model from omitting unfavorable locations due to random deployment.

The example scenarios in Figure 10 also help illustrate the applicability of a maximum flow problem formulation. Although omitted for clarity, the network formulation contains a source node and a directed edge from the source node to every target node. By setting the capacity on each of these edges to one, every unit of flow that reaches the sink must travel from the source node to a different target node, and then through the network to the sink node. Therefore, by determining the number of sensors deployed in each subregion to maximize the average flow from the source to the sink node, the expected coverage of the WSN deployed based on this policy is also maximized.

5.2.2 Stochastic Optimization Model

For a set of scenarios $\Omega = \{1, 2, \ldots, W\}$, the stochastic optimization model is now formulated below in (33)$-$(44). The decision variables in the model include the number of sensors deployed in a subregion, $n_r, r \in \mathcal{R}_s$, the binary decision variable indicating if sensor i is deployed or not, $\gamma(i), i \in \mathcal{N}_1$, the resulting network coverage (or flow) for a scenario, $\beta^\omega, \omega \in \Omega$, and the flow variables, $f^\omega(i,j), (i,j) \in \mathcal{A}^\omega$, necessary that require a sensor within range of a target node and a communication path to the sink node for a target to be covered in a scenario.

The objective function in (33) maximizes the average WSN coverage over all scenarios. Constraint (34) requires that the number of sensors placed in the network equal the total number n, while Constraint (35) requires that the sum of the binary decision variables for every sensor potentially deployed in a subregion equal the number of sensors deployed in the subregion. Constraint (35) also ensures the number of sensors deployed in a subregion does not exceed the maximum number N. Constraint (36) enforces the randomness in sensor location as previously described by requiring that a sensor i can only be deploy if sensor $i-1$ is also deployed. Constraints (37) and (38) enforce an integer restriction on the number of sensors deployed to a subregion and the binary decision variable for every potential sensor deployed, respectively.

$$\max \frac{1}{W} \sum_{\omega=1}^{W} \beta^{\omega} \tag{33}$$

Subject to

$$\sum_{r \in \mathcal{R}_s} n_r = n, \tag{34}$$

$$\sum_{i=1}^{N} \gamma((r-1)N+i) = n_r, \qquad \text{for } r \in \mathcal{R}_s, \tag{35}$$

$$\gamma((r-1)N+i+1) \leq \gamma((r-1)N+i), \qquad \text{for } i = 1, 2, \ldots, N-1, r \in \mathcal{R}_s, \tag{36}$$

$$n_r \in \{0 \cup \mathbb{Z}^+\}, \qquad \text{for } r \in \mathcal{R}_s, \tag{37}$$

$$\gamma(i) \in \{0, 1\}, \qquad \text{for } i \in \mathcal{N}_1, \tag{38}$$

$$\sum_{j:(i,j) \in \mathcal{A}_4} f^{\omega}(i,j) = \beta^{\omega}, \qquad \text{for } i \in \{0'\}, \omega \in \Omega, \tag{39}$$

$$\sum_{j:(i,j) \in \mathcal{A}^{\omega}} f^{\omega}(i,j) - \sum_{j:(j,i) \in \mathcal{A}^{\omega}} f^{\omega}(j,i) = 0, \qquad \text{for } i \in \{\mathcal{N}_1 \cup \mathcal{N}_2\}, \omega \in \Omega, \tag{40}$$

$$\sum_{j:(j,i) \in \mathcal{A}_1^{\omega}} f^{\omega}(j,i) = \beta^{\omega}, \qquad \text{for } i \in \{0\}, \omega \in \Omega, \tag{41}$$

$$f^{\omega}(i,j) \leq 1, \qquad \text{for } (i,j) \in \{\mathcal{A}_3^{\omega} \cup \mathcal{A}_4\}, \omega \in \Omega, \tag{42}$$

$$f^{\omega}(i,j) \leq m\gamma(i), \qquad \text{for } (i,j) \in \{\mathcal{A}_1^{\omega} \cup \mathcal{A}_2^{\omega}\}, \omega \in \Omega, \tag{43}$$

$$f^{\omega}(i,j) \geq 0, \qquad \text{for } (i,j) \in \mathcal{A}^{\omega}, \omega \in \Omega. \tag{44}$$

Constraints (39)−(44) capture the flow balance constraints common to a maximum flow problem, which model the coverage achieved by a sensor deployment in our model. From our construction of the network, flow travels through the network from the source node through a target node, then through a number of sensor nodes as necessary to reach the sink node.

By limiting the capacity of every source-to-target arc to a unit of flow in Constraint (42), every unit of flow that reaches the sink node corresponds to a different target that is covered in the WSN. Therefore, the total value of the flow equals the coverage of the WSN, and can also be bounded by the number of targets m. Constraint (43) limits the flow along a sensor-to-sensor arc (i, j) or sensor-to-sink arc $(i, 0)$, where the flow can only be positive if sensor i is deployed in the network. If the sensor is not deployed then the capacity of the edge, along with every other outgoing edge) is zero, and due to the flow balance constraint in (40) the flow into the sensor node must also be equal to zero. The capacity and flow balance constraints require that a target can only be covered if it is within range of a sensor deployed in the network, and this sensor is able to communicate with the sink node.

5.3 Solution Methodology

For a set of scenarios generated, the model formulation in (33)$-$(44) is a mixed-integer linear program. The first solution approach we consider is an exact method by solving the model using CPLEX. Results are presented when a small number of scenarios are generated, which indicate CPLEX is not well suited to address the large number of scenarios necessary to accurately capture the randomness in sensor locations. This motivates our heuristic search method, which we outline before discussing results on a wide range of test instances.

5.3.1 Test Instance Parameters

As a region of interest we consider a $[0, 1] \times [0, 1]$ unit square, partitioned into $R = 16$ subregions of equal size 0.25×0.25. WSN coverage is determined by the number of targets covered, where $m = 441$ target nodes are uniformly spaced over $\mathcal{R}$ as a 21×21 grid. The number of sensors deployed in the network may vary based on the test instance, however sensor capabilities are constant throughout with a communication radius of $d_1 = 0.075$ and a monitoring radius of $d_2 = 0.075$. The sink node is deployed centrally in the network, but can easily be modeled by a random location as well.

5.3.2 Exact Approach

Scenarios are randomly created in c++ and the model is solved using IBM ILOG CPLEX Optimization Studio version 12.8. Due to the complexity commonly encountered in location/covering problems a direct CPLEX implementation may only be able to handle a relatively small number of scenarios [28]. As Table 7 illustrates, we attempt to solve the model exactly when $W = 10$ and $W = 20$ scenarios are generated, and set a time limit of 12 hours and 24 hours, respectively.

Table 7 also presents results related to each test instance. The "Expected Coverage" column contains the maximum expected coverage (i.e., objective function (33)) resulting from the corresponding deployment policy at termination. The following column, "Computation Time," provides the computation time CPLEX required before returning a solution and corresponding objective function value. Note that in every test instance CPLEX was unable to terminate with an optimal solution and instead reached the time limit. As a result, the "Best Bound" column reports an upper bound on the objective function with the optimality gap between the upper bound and best solution found at termination in the final column.

The primary result of interest from Table 7 is the intractability of a direct CPLEX implementation, especially for a larger number of scenarios. While variation in computation time can be expected when only a small number of scenarios are generated, it is reasonable to

Table 7: CPLEX Implementation

n	W	Expected Coverage (%)	Computation Time (sec)	Best Bound (%)	% Gap
304	10	95.49	43200	97.53	2.09
304	20	91.93	86400	96.60	4.84
288	10	91.32	43200	96.52	5.40
288	20	86.79	86400	96.40	9.97
272	10	86.49	43200	95.24	9.18
272	20	79.19	86400	95.77	17.31

expect that even if an optimal solution is found prior to reaching the time limit a significant amount of time will still be required. Further, decreasing the number of sensors available to deploy corresponds to a more constrained network (e.g., harder to cover the entire region) and increasing the number of scenarios increases both the number of decision variables and the number of constraints in the model. Not surprisingly, as the test instances become more difficult the optimality gap of CPLEX at termination also increases. When the model is solved with only a small number of scenarios there is a larger variability in the objective function value, and as we describe below the optimal solution is dependent more on the actual scenarios sampled. As a result the solution quality is not as large a concern, but the computation time is still of interest. With this insight and a desire to solve the model when a much larger number of scenarios are generated we explore a heuristic method to determine an optimal deployment policy.

5.3.3 Heuristic Approach

Similar to [12–15] we explore a heuristic method to determine a deployment policy that maximizes expected coverage. We are interested in a heuristic solution method capable of identifying high quality solutions while not requiring the significant computation time required by an exact approach, allowing the heuristic to solve the model for a much larger number of scenarios. With this direction we propose a heuristic search method comprised of two main steps. The first is a greedy search method which provides a feasible solution to the problem, followed by a neighborhood search designed to look for solutions that improve expected coverage. Both steps leverage the observation that for a given deployment solution (i.e., how many sensors are deployed in each subregion) expected coverage can be estimated with a breadth-first search (BFS) algorithm implemented on each scenario. The significance of this observation is that a BFS algorithm on a modified network (where the direction of edges in $\mathcal{A}_2^\omega$ and $\mathcal{A}_3^\omega$ are reversed) is more efficient than using a maximum flow algorithm to determine the resulting coverage in a scenario. A maximum flow formulation is necessary for the optimization model, but if a deployment solution is provided than expected coverage can be estimated in a more efficient manner.

The greedy search method starts from a solution in which N sensors are deployed in every subregion of the network. This is likely an infeasible solution since we are only able to deploy n sensors, but it provides an upper bound on the expected coverage for a network consisting of n sensors. The greedy method proceeds by iteratively removing q_1 sensors from the network until a feasible solution is reached. From the current solution, the expected coverage resulting from the removal of q_1 sensors from a subregion is estimated by temporarily removing q_1 sensors from the subregion and applying a BFS to every scenario. Repeating this process for every subregion allows the subregion that results in the smallest decline in expected coverage to be selected, and the current solution is then updated by the removal of q_1 sensors from the corresponding subregion.

Once a feasible solution is reached (i.e., a solution that deploys exactly n sensors), a neighborhood search algorithm is applied in an effort to improve the current deployment policy. This is accomplished by methodically moving a small number of sensors from one subregion to another, maintaining a feasible solution, and determining which relocation, if any, results in an improvement to expected coverage. Since we are removing sensors from one subregion we anticipate this decreasing network coverage, however this allows a larger number of sensors to be deployed in another subregion which may increase the overall coverage compared to the initial deployment. The neighborhood of a current solution is generated by moving $q_2 \geq 1$ sensors from some subregion i to another subregion j, for all subregions with at least one sensor. For example, consider the region $\mathcal{R}$ partitioned into three subregions, with $4, 6,$ and 5 sensors deployed in each subregion, respectively. The current solution is represented by $[n_1\ n_2\ n_3] = [4\ 6\ 5]$, and if $q_2 = 2$ the neighborhood is

$$\text{neighbors}([4\ 6\ 5]) = \begin{bmatrix} 2 & 8 & 5 \\ 2 & 6 & 7 \\ 6 & 4 & 5 \\ 4 & 4 & 7 \\ 6 & 6 & 3 \\ 4 & 8 & 3 \end{bmatrix}.$$

The expected coverage for each of the neighborhood solutions is estimated, and the solution resulting in the maximum expected coverage is selected to update as the current solution. This process repeats until the current solution is the best compared to those evaluated in the neighborhood (i.e., a local optima based on neighborhood definition). One appealing aspect of the proposed heuristic approach is the simplicity with which it can be implemented. Both the greedy removal of sensors and subsequent neighborhood search phase rely on an implementation of a BFS algorithm, which can be accomplished in $O(V+E)$ time on a network with V nodes and E edges. Therefore, increasing the number of scenarios corresponds to an increase in the number of BFS algorithms implemented each iteration of the heuristic, and the low computational complexity support the belief that the heuristic should be able to address a larger number of scenarios without the significant increase in computation time.

Table 8 restates a subset of the results from the CPLEX implementation for each test instance, along with the expected coverage and computation time required by the heuristic solution method on the same instances generated. This allows for a more direct comparison between the expected coverage and computation time of the two methods. The heuristic approach requires significantly less computation time than the exact approach. Whereas CPLEX reached the maximum time limit of 12 and 24 hours, the heuristic returned a solution to each instances in a few seconds. Additionally, the increase in computation time required by doubling the number of scenarios matches that expected based on the previous discussion.

Table 8: Heuristic Comparison

		Exact Solution		Heuristic Solution	
n	W	Expected Coverage (%)	Computation Time (sec)	Expected Coverage (%)	Computation Time (sec)
304	10	95.49	86400	93.54	5.56
304	20	91.93	86400	90.56	13.66
288	10	91.32	43200	87.10	6.73
288	20	86.79	86400	85.58	13.68
272	10	86.49	43200	84.58	6.08
272	20	79.19	86400	80.86	13.86

In comparing the expected coverage of the heuristic and CPLEX solution, it is worth mentioning that the expected coverage for a solution based on such a small number of scenarios can be misleading. With a limited number of scenarios there is more opportunity to tailor a solution based on common features between scenarios (e.g., the ith sensor is always sampled to land at a favorable location), which will not be present when a larger number of scenarios are generated. That is, it is difficult to accurately capture the randomness of a sensor randomly deployed with a small number of scenarios. As a result, expected coverage is influenced more by the actual scenarios generated which leads to an overestimate on expected coverage compared to the average performance of a solution over a much larger number of scenarios.

With this in hand, conclusions based on the expected coverage of the heuristic solution compared to the best CPLEX solution returned are not entirely appropriate. An exact solution approach is better suited to identify similarities between a small number of scenarios, resulting in policies more dependent on the actual scenarios generated. The heuristic approach might have a more difficult time identifying similar structure between scenarios, but it is not designed to do so. The heuristic is intended to be applied to a much larger number of scenarios that better capture the randomness in sensor location and the probability of similarities among every scenario influencing the policy is less likely. Therefore, while it is appealing that the expected coverage of the heuristic solution is comparable to the CPLEX solution, the fact that the optimality gap of the heuristic solution in the smaller test instances is up to 15% is not a cause of concern. The primary value of the results provided in Table 8 is the drastic reduction in computation time required by the heuristic, and its potential to solve the model with a much larger number of scenarios. In the following section we discuss results related to the quality of the expected coverage a heuristic solution achieves by comparing the expected coverage to a baseline uniform policy.

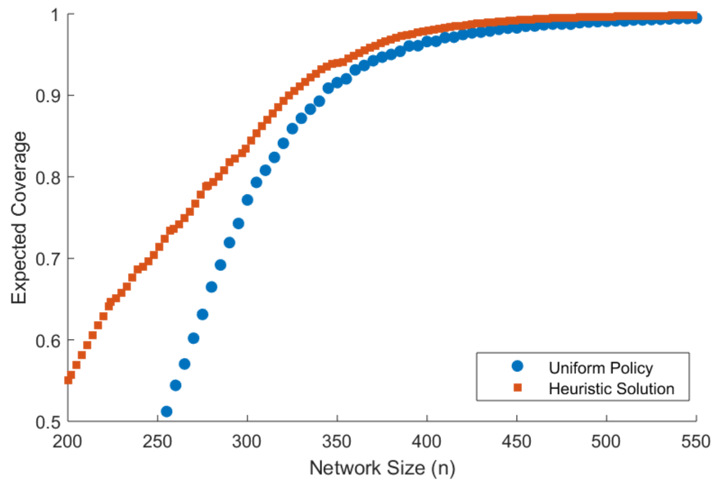

Figure 11: Heuristic Solution Comparison with Uniform Distribution of Sensors

5.4 Computational Study and Model Extensions

The heuristic solution method now provides the desired capability of solving the model with a much larger number of scenarios, the results of which are discussed in this section. For a study on the performance of the heuristic solution, the number of sensors deployed in the network varies from $n = 550$ down to $n = 200$. The region of interest, partition into subregions, target distribution, and sensor capabilities are identical to those introduced in Section 5.3.1. The number of sensors removed during the greedy search phase of the heuristic is set by $q_1 = 3$, and neighborhood of the current solution is defined by relocating $q_2 = 2$ sensors from one subregion to another. Figure 11 presents the expected network coverage for different network sizes, where $W = 1,000$ scenarios are generated are generated for every test instance.

As previously discussed, when sensors are randomly deployed in a WSN it is common to assume they are randomly deployed with uniform density throughout the region of interest. Therefore, the expected coverage of the heuristic policy is compared to a uniform policy in which n sensors are randomly located over $\mathcal{R}$. This is equivalent to randomly selecting a subregion, increasing the number of sensors deployed in the subregion by one, and repeating until all n sensors have been deployed. While policies that randomly deploy sensors according to a non-uniform distribution (e.g., simple diffusion, R-random, etc.) have been proposed, the computation study in [23] suggests that a uniform distribution of sensors is one of the best policies with respect to coverage.

A comparison of the expected coverage resulting from the heuristic solution and the uniform policy is illustrated in Figure 11. For networks with a large number of sensors offering near perfect target coverage there is less opportunity for the heuristic solution to reallocate sensors and improve WSN performance. The improvement in expected coverage is more noticeable for smaller networks when the WSN struggles to cover the entire region. For example, when 300 sensors are available to deploy in the network a random deployment strategy over the entire region results in an expected coverage of approximately 77%. Comparatively, the heuristic solution achieves an expected coverage of approximately 84%, significantly improving upon a random deployment strategy. An alternative comparison is the number of sensors each policy requires to achieve a given expected coverage requirement. For example, consider the task of determining the number of sensors necessary such that the network covers 90% of the region. The random distribution strategy requires approximately 350 sensors, compared to the heuristic solution which achieves the same coverage with only 325 sensors. In situations where a smaller network size is desirable, possibly due to cost or ease of network deployment, the heuristic solution allows a smaller network to satisfy the given coverage requirement compared to a random deployment strategy.

As the number of sensors available to deploy decreases the improvement in expected coverage achieved by the heuristic solution becomes more profound. For the smaller network sizes with $n \leq 275$ this improvement is attributed to the heuristic solution sacrificing network coverage in one subregion in order to deploy a larger number of sensors in the remaining subregions. That is, by deploying zero sensors in one subregion the network provides no coverage in this portion of the region. However, sensors can now be deployed in a grater density in the remaining subregions, which improves the ability of sensors to communicate and increases the probability a sensor lands within monitoring distance of a target. Meanwhile, randomly deploying the same number of sensors over the entire region of interest results in a large number of sensors isolated from the sink node, and sensors struggle to route information through the network. While the test instances this phenomena occur in might not appear as interesting since it is reasonable to question the suitability of a deployment policy, and associated network size, that offers zero coverage in a portion of the region, this is only observed in a small number of the overall instances. In a majority of the test instance (e.g., $n > 275$), the improvement in expected coverage achieved by the heuristic solution is attribute to allocating sensors in an efficient manner to different subregions contributing towards both an improvement in coverage and sensor communication ability over the entire region of interest.

5.4.1 Equal Distribution Policy

A notable difference between the heuristic policy and uniform policy is that the heuristic policy allows more control over how sensors are distributed throughout the network. Perhaps a more appropriate comparison than with a uniform distribution over the entire region is a policy that evenly distributes sensors to each subregion in the network. This is referred

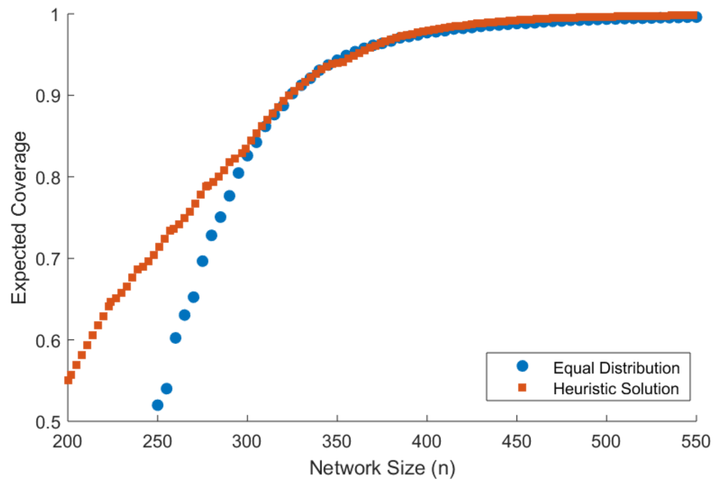

Figure 12: Heuristic Solution Comparison with Equal Distribution of Sensors to Every Subregion

to as an equal distribution policy, where n sensors are allocated so each subregion contains approximately $\lfloor n/n_r \rfloor$ sensors. The uniform policy is similar to the equal distribution policy in that the expected number of sensors that land in each subregion in the uniform policy is n/n_r. However the observed number of sensors that land in a subergion will vary due to randomly deploying sensors over the entire region, compared to the equal distribution policy which removes this source of variation.

A comparison of the heuristic solution with the equal distribution policy is illustrated in Figure 12. Once the number of sensors deployed in the network is above 300, there appears to little or no improvement offered by the heuristic policy. For the smaller sized instances this is again attributed to the heuristic solution sacrificing coverage in at least one subregion to improve coverage in the remaining subregions. This leaves only a small handful of instances where the heuristic policy improves upon the equal distribution policy and offers coverage throughout the entire region before the performance of the two policies is almost identical. Therefore, it might be reasonable to always select the equal distribution policy and avoid solving the proposed stochastic optimization model. In the following section we discuss an extension of the modeling approach when sensors are prone to failure, and provide results indicating the heuristic solution to the stochastic optimization model with sensor failures achieves a larger expected coverage than an equal distribution policy.

5.4.2 Homogeneous Sensor Failures

In addition to communication and monitoring components, sensors contain a finite power supply enabling sensor operations [29]. It is important for a WSN to not only satisfy a coverage requirement when it is initially deployed, but also over a period of time and tolerate sensor failures without a large drop in WSN capability. Sensor failures influence the ability of sensors to route information through the network and target coverage, which might impact the initial deployment decision. In this section we discuss a slight modification enabling the stochastic optimization model to determine an optimal deployment policy in the presence of sensor failures. In fact, the model formulation in (33)− (44) does not actually change; the modification is in constructing the network for a given scenario.

To model sensor failures we assume that each sensor deployed in the network fails with some probability p. Similar to the problem formulation with no failures, for every scenario a random location (x_i^ω, y_i^ω) is sampled in the appropriate subregion for each potential sensor $i \in \mathcal{N}_1$. In addition to a random location, an indicator variable, z_i^ω, is sampled to reflect if sensor i is functioning or failed in the scenario. z_i^ω follows a Bernoulli distribution, equal to zero (i.e., a failed sensor) with probability p, and one (i.e., a functioning sensor) with probability $1 - p$. Now, when creating the edge sets $\mathcal{A}_1^\omega, \mathcal{A}_2^\omega$, and $\mathcal{A}_3^\omega$, the status of each sensor is also considered. That is, for any edge originating and/or ending at a sensor node,

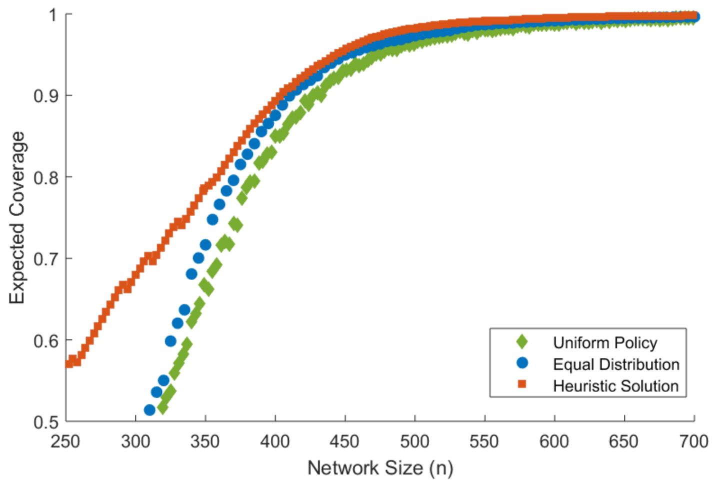

Figure 13: Solution Comparison With Probability of Sensor Failure $p = 0.2$

the edge is only present if the corresponding sensor is functioning. This equates to edge sets defined by $\mathcal{A}_1^\omega = \{(i,j) : d^\omega(i,j) \leq d_1, z_i^\omega = z_j^\omega = 1, i \in \mathcal{N}_1, j \in \mathcal{N}_1, j \neq i\}$, $\mathcal{A}_2^\omega = \{(i,0) : d^\omega(i,0) \leq d_1, z_i^\omega = 1, i \in \mathcal{N}_1\}$, and $\mathcal{A}_3^\omega = \{(i,j) : d^\omega(i,j) \leq d_2, z_j^\omega = 1, i \in \mathcal{N}_2, j \in \mathcal{N}_1\}$. The edge set $\mathcal{A}_4$ is unchanged, resulting in the overall edge set $\mathcal{A}$ consisting of edges between the sink node, target nodes, and functioning sensor nodes.

With a modification on the network construction for each scenario, the resulting optimization model and solution methodology remain unchanged. The expected coverage resulting from each of the three policies for a sensor probability of failure of $p = 0.2, p = 0.3,$ and $p = 0.4$ is provided in Figure 13 – Figure 15, respectively. Compared to the initial model with perfectly reliable sensors (i.e., no failures), the heuristic solution to the stochastic optimization model including failures is able to improve upon the equal distribution of sensors and provides an increase in expected coverage.

For the larger sized networks sensors can be deployed to withstand failures and provide coverage over a significant portion of the region. Since larger sized networks are more robust to sensor failure and any decrease in target coverage will be minor, Figure 12 suggests that

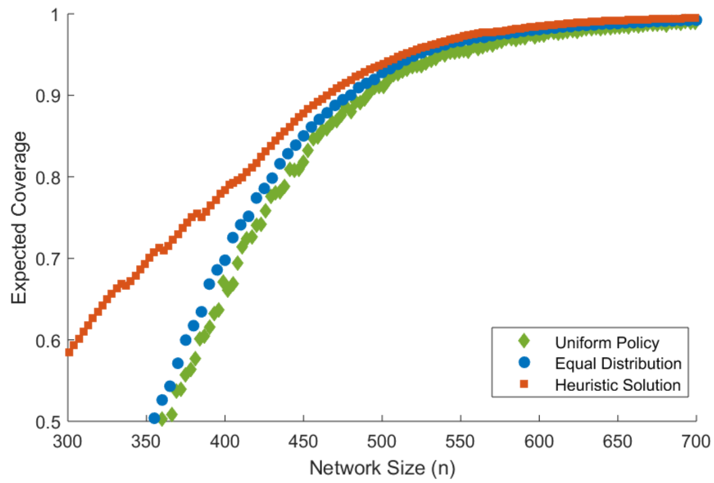

Figure 14: Solution Comparison With Probability of Sensor Failure $p = 0.3$

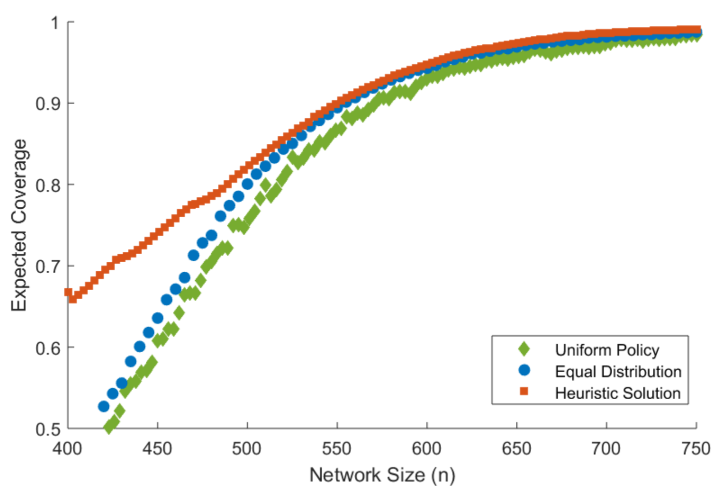

Figure 15: Solution Comparison With Probability of Sensor Failure $p = 0.4$

the expected coverage achieved by the heuristic policy and an equal distribution policy will again be similar. This is indeed observed in Figure 13 – Figure 15, where once the expected coverage of an equal distribution policy reaches 90% any improvement achieved by the heuristic policy is relatively minor. While there is some benefit to the heuristic policy in these test instances, for each of the three failure probabilities considered the increase in expected coverage over an equal distribution policy is around 0.5%, on average.

For smaller sized networks the impact of a sensor failure is more profound and an equal distribution policy might not be well suited to maintain a communication path to the sink node and/or provide redundant sensors monitoring a target. While deviating from an equal distribution policy (e.g., to the heuristic policy) will not improve these conditions throughout the entire network, there might be a few critical locations where deploying a larger number of sensors will increase overall network capability. The improvement achieved by the heuristic policy is more noticeable as the network size decreases, but is best illustrated when the resulting network coverage is between 75% and 90%. For each of the failure probabilities evaluated, the improvement achieved by the heuristic policy does not require sacrificing coverage in one or more subregions. That is, similar to an equal distribution policy the heuristic solution provides coverage over the entire region of interest, but deploys a varying number of sensors in different subregions to maintain a communication path and/or coverage of a target as sensors fail.

The heuristic policy achieves this increase in expected coverage primarily by deploying a larger number of sensors in subregions immediately surrounding the sink node. Since a sensor must also be able to communicate with the sink node, a sensor that monitors a large number of targets but is disconnected from the sink node provides no capability to the overall network. The heuristic policy deploys a slightly larger number of sensors in the subregions surrounding the sink node to maintain a communication path and reduce the number of sensors that become disconnected from the sink node when a sensor fails. Considering only the resulting heuristic solutions that deploy a positive number of sensors in each subregion to provide coverage throughout the entire region, the increase in expected coverage over an equal distribution policy is around 3.5% on average, for each of the failure probabilities considered. The improvement of the heuristic policy is most significant with a sensor failure probability of $p = 0.3$, where the increase in expected coverage over an equal distribution policy is up to 8%.

For ease of illustration each sensor was assumed to have a constant probability of failure p. A more interesting approach might be to model sensor failures over time, where $T \geq 0$ represents the lifetime of a sensor and is modeled by a cumulative distribution function $F(t)$. For example, say we are interested in maximizing network coverage at time t, at which new sensors will be deployed to maintain the network over a prolonged period of time. The proposed model is dependent on the time interval t, but is otherwise well suited to address a given sensor failure distribution. The model can also handle multiple failure distributions, for example to reflect the energy hole phenomena in which sensors located around the sink node fail at a faster rate [30], by simply changing the failure probability p for a sensor when sampling the sensor status z_i^ω.

5.4.3 Impact of Sensor Failures on Deployment Policy

We may also be concerned with the influence of sensor failure probability on an optimal deployment policy. For this purpose we first consider the test instance that deploys $n = 500$

sensors in the network, and examine how the deployment policy changes with the probability of sensor failure. Rather than directly compare the number of sensors deployed in each subregion, we first group the subregions based on their relation to the sink node. From Section 5.3.1 the region of interest is partitioned into 16 subregions each of size 0.25×0.25, which can be viewed as a 4×4 grid overlaying the region (compared to the 2×2 grid in Figure 10). Since the sink node is deployed centrally in the network, the four inner subregions (grouped into "center subregions") are all adjacent to the sink node. Meanwhile, there are four "corner subregions" located near the border of the region, with each corner subregion located diagonally from one of the center subregion. The remaining eight "border subregions" are adjacent to a single edge of the border in the overall region. Based on this grouping, the corner subregion group and the center subregion group each comprise 25% of the overall region, and the border subregion group the remaining 50%.

Table 9 presents the expected coverage and the percent of sensors deployed in each subregion group from the heuristic policy for each of the three failure probabilities considered when 500 sensors are deployed in the network. As expected, increasing the probability of sensor failure lowers the expected coverage of the resulting WSN. It is more interesting to observe how the failure probability affects the distribution of sensors in the network. Notice that the smallest failure probability results in a policy that deploys a larger percentage of sensors in the four corner subregions than in the four center subregions. This can be attributed to a bounded region of interest and a characteristic of the corner subregions, and to a smaller extent the border subregions, that is not present in the center subregions. When a sensor is deployed to a corner subregion there is a possibility it lands near the border of the region. This is desirable since there are targets near the border for the sensor to monitor.

However the closer the sensor lands to the border, the sensor is now only able to communicate with neighbor sensors that are located closer to the interior of the subregion. As a result any portion of the communication area that falls outside the region is lost or unusable, and there is smaller effective area for neighboring sensors to land in necessary for connectivity to the sink node. Similarly, for targets located near the border there is a smaller area of the region a sensor must land in for the target to be monitored compared to a target located closer to the sink node. Therefore, it might be necessary to deploy a larger number of sensors to a corner or border subregion to increase the probability a sensor lands within monitoring distance of a target on the border of $\mathcal{R}$ and ensure the sensor is connected to the sink node.

Table 9: Policy Comparison When Deploying $n = 500$ sensors

p	Expected Coverage	Corner Subregions	Border Subregions	Center Subregions
0.2	98.25%	29.87%	47.75%	22.38%
0.3	93.77%	26.82%	47.25%	25.93%
0.4	82.04%	21.52%	49.85%	28.63%

When the probability of an individual sensor failure is relatively low, the decline in network connectivity is not as severe and a larger focus can be dedicated to coverage. This helps explain why a larger proportion of sensors are deployed in the corner subregions than the center subregions when $p = 0.2$. As the failure probability increases maintaining network connectivity becomes more of a priority. To ensure sensors remain connected to the sink node in the presence of failures, it becomes desirable to deploy a larger number of sensors surrounding the sink node. This is observed with a failure probability of 0.3 and 0.4, where the proportion of sensors deployed in the corner subregions starts to decline and a larger proportion of sensors are now deployed in the center subregions.

In Table 10 we allow the total number of sensors deployed to vary based on the failure probability such that the heuristic policy achieves a selected expected coverage level. A notable observation from Table 10 is that for a given coverage level, the proportion of sensors deployed in each of the three subregion groups is approximately the same for the different failure probabilities. For example, in each of the deployment policies resulting in an expected coverage of 90% based on the network size and failure probability combination, approximately 24% of the sensors are deployed in corner subregions, 26% in the center subregions, and the remaining 49% of sensors in border subregions. While there is slightly more variability as the expected coverage decreases, the proportion of sensors deployed in each group remains relatively similar. Particularly for the center subregions, which contain approximately 29.5% of the sensors and 34% of the sensors to achieve an expected coverage of 80% and 70%, respectively.

Table 10: Policy Comparison For Selected Coverage Levels

Expected Coverage	p	n	Corner Subregions	Border Subregions	Center Subregions
90%	0.2	405	23.98%	49.81%	26.21%
90%	0.3	465	23.73%	49.95%	26.32%
90%	0.4	550	24.43%	49.05%	26.52%
80%	0.2	360	20.72%	49.79%	29.49%
80%	0.3	410	19.03%	51.39%	29.58%
80%	0.4	490	20.82%	49.85%	29.33%
70%	0.2	315	10.05%	56.14%	33.81%
70%	0.3	360	8.59%	57.06%	34.35%
70%	0.4	425	11.75%	54.41%	33.84%

Similar to Table 9, Table 10 also helps illustrate how a policy changes for a fixed failure probability based on the number of sensors deployed. When both the failure probability p

and number of sensors deployed n differ, comparing policies with respect to the proportion of sensors in each subregion grouping can be informative. It is also valuable to consider the change in the actual number of sensors deployed in each subregion group, particularly when the failure probability is fixed. While not directly provided, this information is obtainable from Table 10. In comparing the number of sensors deployed in each subregion group, it is interesting to observe that for a fixed failure probability the total number of sensors deployed in the center subregions remains unchanged for the different values of n illustrated in Table 10. That is, in each instance with $p = 0.2$ approximately 106 sensors are deployed in the center subregions, with $p = 0.3$ approximately 122 sensors are deployed in the center subregions, and with $p = 0.4$ approximately 144 sensors are deployed in the center subregions. Therefore, even though the number of sensors deployed in the center subregions comprise a larger proportion of the deployment policy, the number of sensors deployed in these subregions is unchanged. This indicates that a primary difference in the deployment policy as n decreases is that a smaller number number of sensors are deployed in the corner subregions, and to a smaller extend the border subregions as well. When considering the larger failure probability of $p = 0.4$ and the decrease in n from 550 to 425, this result is somewhat surprising since there are 125 fewer sensors available to deploy but the policy still deploys the same number of sensors in the center subregions while deploying a significantly smaller number of sensors to the corner subregions.

5.4.4 Heterogeneous Sensor Failures

From the discussion at the end of Section 5.4.2 the proposed model offers the flexibility to address heterogeneous sensor failure probabilities as well. The energy hole problem is a common issue in WSNs in which sensors near the sink node fail at a faster rate since they are relied upon more often to route information [25, 30, 31]. One approach to reflect a larger use of sensor nodes close to the sink is to set a failure probability, or failure rate in a lifetime distribution $F(t)$, based on the distance of the sensor to the sink node for a given scenario. To mitigate the energy hole problem it may be possible to aggregate information at various sensor nodes to reduce the total number of messages transmitted [32] or place a subset of nodes in an energy conserving state, switching to an active state later on when necessary [33]. Similar to [25] and [26] nonuniform deployment policies related to the energy hole problem commonly incorporate a lifetime objective, for example maximizing the time of the first sensor failure.

The strategies discussed in [22] and [23] are also well suited to address the energy hole problem and provide a more detailed discussion related to the coverage of a given policy. To model the failure of sensors over time, each sensor has an individual starting power supply that is consumed by sending and receiving messages. Models related to energy consumption, and therefore sensor failure, can be complicated by a number of features such as whether or not sensors fluctuate between an active or inactive state [33], the transmission power/radius of a sensor [34], the path selected to route information to the sink node [1], and the rate at which sensors transmit information [35]. In this work we select a simpler approach by modeling sensor failures based on a probability, which avoids the numerous assumptions necessary for an energy consumption model. It is also uncommon for energy consumption models to address random sensor failures, for example arising when sensors are deployed in a harsh environment, and suffer from physical damage [2], or software malfunctions [36].

Based on the previous discussion and a model focused on maximizing network coverage rather than a cost or lifetime objective, we do not believe a direct comparison with similar nonuniform deployment strategies such as those discussed in [22, 23, 25] is entirely appropriate. Policies concerned more with energy consumption tend to place a larger emphasis on connectivity and minimizing isolated sensors, which may impact coverage, rather than the trade-off between increasing connectivity near the sink node and coverage in remaining regions of the network. The purpose of this section is rather to discuss a simple extension of our proposed model capable of addressing multiple failure probabilities and an example of where such a situation might be encountered.

There is one component to the solution of our model not present in a fixed policy (such as simple diffusion, R-random, etc.) worth highlighting. Demonstrated in Table 10, the proportion of sensors in a given subregion group differs based on both the network size and sensor failure probability. As network size increases and the number of sensors deployed near the sink node reaches a sufficient level to maintain connectivity, the optimal policy starts to focus on subregions farther away from the sink node. Policies reliant on a fixed distribution function, without considering network size and failure probability, may place a greater focus on subregions near the sink node than necessary. For example if there is only a small probability of sensor failure than the density of sensors near the sink node may only need to be increased slightly, and the number of sensors deployed near the border of the region of interest can remain relatively high. Similarly, if there is a high probability of sensor failure but there are a large number of sensors available to deploy as well than the overall focus might remain on improving coverage in the border regions. In a deployment policy with a fixed distribution function increasing the number of sensors deployed likely results in a larger number of sensors deployed near the sink node with only a small number of sensors in the exterior regions where an additional sensor will actually offer a larger improvement to coverage and connectivity. Based on the results in Table 10 parameters such as the network size and failure probability should influence a deployment policy as well. Capturing a similar behavior when deploying sensors according to a fixed distribution function might be possible by varying distribution parameters (e.g., variance in distance from sink, center location), mentioned in [25], or the distribution function itself (e.g., normal, uniform, simple diffusion) based on the network size and failure probability, but has not been widely investigated and is more commonly modeled by selecting a single distribution function.

5.5 Conclusion

Depending on the application and operating environment of a WSN, sensors might be placed at deterministic locations or randomly deployed throughout a region of interest. The initial deployment of a WSN can have a significant influence on the ability of sensors to communicate with one another and the resulting coverage achieved by the network. Related research has focused primarily on deploying a fixed number of sensors in a deterministic setting to maximize coverage [12], the smallest number of uniformly deployed sensors to satisfy full coverage [21], or analyzing the performance of a few nonuniform random deployment policies [22, 23]. Nonuniform deployment policies are also common in addressing the energy hole issue in which sensors are deployed in a greater density near the sink node to offset the increased failure rate [25].

In this work we focus on maximizing expected coverage when there is uncertainty in where a sensor lands. While problems that maximize a measure of network lifetime are similar, they are more commonly defined in terms of the first sensor failure or the time at which a sensor is isolated from the sink node. The main contribution of this work is a stochastic optimization model, formulated as a collection of maximum flow problems, to determine a deployment policy that maximizes expected coverage. The model is of particular interest when full region coverage is not attainable but the resulting network is still able to cover a large portion of the region. To solve the model when a large number of scenarios are generated, which is necessary for an accurate estimate on the expected coverage of a solution, we present a heuristic solution approach that determines the number of sensors deployed in various subregions of the network. We also discuss slight modifications allowing the model to handle both homogeneous and heterogeneous sensor failures, and present results for a number of fixed failure probability test instances. The performance of our heuristic solution is compared to the more common assumption that sensors are uniformly distributed throughout the region and a more controlled equal distribution policy. As the number of deployed sensors decreases and/or there is a increased probability of sensor failure, the heuristic solution efficiently allocates sensors in different subregions of the network to improve expected coverage over both the uniform and equal distribution policy.

The stochastic optimization model also provides several directions to build on in future work. An appealing aspect of the proposed heuristic is the low computation complexity and ease of implementation. There are several opportunities to explore different heuristic approaches to solve for a deploy policy, which might result in a policy that further improves expected coverage. For example, one option is a more complex neighborhood definition or an improved search in the solution space.

It might also be possible to deploy a number of relay sensors throughout the network to assist network communication. While relay sensors are typically more powerful and have a larger communication radius, they are also more expensive than individual sensor nodes [7]. If a small number of relay sensors are available to deploy as well, it is interesting to see how a deployment policy utilizes these relay sensors (e.g., where they are deployed in the network), and the influence on the current deployment policy.

References

[1] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society*, vol. 3. IEEE, 2001, pp. 1380–1387.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, 2002.

[3] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2006, pp. 131–142.

[4] K. Kar and S. Banerjee, "Node placement for connected coverage in sensor networks," in *WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.

[5] K. Islam, W. Shen, and X. Wang, "Wireless sensor network reliability and security in factory automation: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1243–1256, 2012.

[6] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Networks*, vol. 14, no. 3, pp. 347–355, 2008.

[7] E. L. Lloyd and G. Xue, "Relay node placement in wireless sensor networks," *IEEE Transactions on Computers*, vol. 56, no. 1, pp. 134–138, 2006.

[8] H. M. Almasaeid and A. E. Kamal, "On the minimum k-connectivity repair in wireless sensor networks," in *2009 IEEE International Conference on Communications*. IEEE, 2009, pp. 1–5.

[9] J. L. Bredin, E. D. Demaine, M. Hajiaghayi, and D. Rus, "Deploying sensor networks with guaranteed capacity and fault tolerance," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2005, pp. 309–319.

[10] P. Szczytowski, A. Khelil, and N. Suri, "Dkm: Distributed k-connectivity maintenance in wireless sensor networks," in *2012 9th Annual Conference on Wireless On-Demand Network Systems and Services*. IEEE, 2012, pp. 83–90.

[11] M. Locatelli and U. Raber, "Packing equal circles in a square: a deterministic global optimization approach," *Discrete Applied Mathematics*, vol. 122, no. 1-3, pp. 139–166, 2002.

[12] Y. Yoon and Y. H. Kim, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1473–1483, 2013.

[13] H. T. T. Binh, N. T. Hanh, N. Dey *et al.*, "Improved cuckoo search and chaotic flower pollination optimization algorithm for maximizing area coverage in wireless sensor networks," *Neural Computing and Applications*, vol. 30, no. 7, pp. 2305–2317, 2018.

[14] N. T. Hanh, H. T. T. Binh, N. X. Hoai, and M. S. Palaniswami, "An efficient genetic algorithm for maximizing area coverage in wireless sensor networks," *Information Sciences*, vol. 488, pp. 58–75, 2019.

[15] D. T. H. Ly, N. T. Hanh, H. T. T. Binh, and N. D. Nghia, "An improved genetic algorithm for maximizing area coverage in wireless sensor networks," in *Proceedings of the Sixth International Symposium on Information and Communication Technology*, 2015, pp. 61–66.

[16] S. Debnath, A. K. Singh, and A. Hossain, "A comprehensive survey of coverage problem and efficient sensor deployment strategies in wireless sensor networks," *Indian Journal of Science and Technology*, vol. 9, no. 45, pp. 1–6, 2016.

[17] L. M. S. De Souza, H. Vogt, and M. Beigl, "A survey on fault tolerance in wireless sensor networks," *Interner Bericht. Fakultät für Informatik, Universität Karlsruhe*, 2007.

[18] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2. IEEE, 2003, pp. 1293–1303.

[19] P. K. Sahoo and J.-P. Sheu, "Limited mobility coverage and connectivity maintenance protocols for wireless sensor networks," *Computer Networks*, vol. 55, no. 13, pp. 2856–2872, 2011.

[20] P.-J. Wan and C.-W. Yi, "Coverage by randomly deployed wireless sensor networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2658–2669, 2006.

[21] B. Liu and D. Towsley, "A study of the coverage of large-scale sensor networks," in *2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems.* IEEE, 2004, pp. 475–483.

[22] M. Ishizuka and M. Aida, "Performance study of node placement in sensor networks," in *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.* IEEE, 2004, pp. 598–603.

[23] M. R. Senouci, A. Mellouk, and A. Aissani, "Random deployment of wireless sensor networks: a survey and approach," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 15, no. 1-3, pp. 133–146, 2014.

[24] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, "Sensor deployment strategy for target detection," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, 2002, pp. 42–48.

[25] S. Halder and A. Ghosal, "Lifetime enhancement of wireless sensor networks by avoiding energy-holes with gaussian distribution," *Telecommunication Systems*, vol. 64, no. 1, pp. 113–133, 2017.

[26] X. Wu, G. Chen, and S. K. Das, "Avoiding energy holes in wireless sensor networks with nonuniform node distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 5, pp. 710–720, 2008.

[27] V. Kapnadak and E. J. Coyle, "Optimal nonuniform deployment of sensors for distributed detection in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 10, no. 2, pp. 1–27, 2014.

[28] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.

[29] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems.* New York, NY: ACM, 2003, pp. 28–39.

[30] J. Li and P. Mohapatra, "An analytical model for the energy hole problem in many-to-one sensor networks," in *IEEE Vehicular Technology Conference*, vol. 62, no. 4, 2005, p. 2721.

[31] P. Szczytowski, A. Khelil, and N. Suri, "Lehp: Localized energy hole profiling in wireless sensor networks," in *The IEEE Symposium on Computers and Communications.* IEEE, 2010, pp. 100–106.

[32] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings 22nd International Conference on Distributed Computing Systems Workshops.* IEEE, 2002, pp. 575–578.

[33] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 1, pp. 89–124, 2005.

[34] M. Li and B. Yang, "A survey on topology issues in wireless sensor network," in *ICWN.* Citeseer, 2006, p. 503.

[35] J. Li and P. Mohapatra, "Analytical modeling and mitigation techniques for the energy hole problem in sensor networks," *Pervasive and Mobile Computing*, vol. 3, no. 3, pp. 233–254, 2007.

[36] C. Vasar, O. Prostean, I. Filip, R. Robu, and D. Popescu, “Markov models for wireless sensor network reliability,” in *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing.* IEEE, 2009, pp. 323–328.

6 Conclusion

Wireless sensor networks (WSNs) are complex structures relying on a large number of sensor nodes to route information through the network and effectively monitor a region of interest. Since WSN performance is heavily influenced by the number of sensors in the network and their locations, a large research effort has focused on methods to extend WSN lifetime such as power management techniques or introducing a small number of new sensors to improve network connectivity. In this work we investigate the maintenance of a WSN over an extended time horizon, in which sensors are deployed in multiple time periods. We present several different models addressing not only the reliability of a deployment policy, whereas similar work has focused on a lifetime or coverage/connectivity measure, but the associated cost of a policy as well.

We first introduce a time-based deployment model in Chapter 2 defined by a constant deployment interval and a fixed network size. Estimating WSN reliability, already a difficult task due to the complex structure of the network, is complicated further by the deployment of new sensors over time. We discuss several characteristics of the network destruction spectrum (D-spectrum) that are well suited to address the complexity of estimating the reliability of a time-based deployment policy. The resulting model formulation allows a wide range of deployment policies to be evaluated in an efficient manner which informs an efficient frontier of deployment policies that balance deployment cost and WSN reliability.

In Chapter 3 we present a myopic condition-based deployment model with a fixed budget and greater control on the deployment of new sensors in the network. As a result, the model must now address the added complexity in that network reliability depends on the deployment action as well. We discuss how the D-spectrum can be incorporated to approximate the reliability of a myopic deployment policy, further demonstrating the value of the D-spectrum in a repeated sensor deployment setting. The significance of the proposed methodology is that the computational effort of a traditional Monte Carlo reliability simulation can be avoided, and a number of deployment policies can efficiently be estimated and compared to one another.

Chapter 4 presents a Markov decision process (MDP) model for the condition-based deployment problem. Compared to the myopic condition-based deployment problem, the MDP model provides a budget for the entire planning horizon and must address the impact of the current decision on future deployment actions as well. We apply approximate dynamic programming (ADP) methodology and an approximate value iteration algorithm to determine an optimal condition-based deployment policy. Our MDP model is one of the few addressing maintenance through the repeated deployment of new sensors in the network, as well as one of the first ADP applications for the maintenance of a complex WSN.

Finally, in Chapter 5 we discuss the problem encountered when first deploying a WSN and how sensors are distributed throughout the region to maximize expected coverage. While there is some control on how sensors are deployed in the region there exists randomness in the exact location of a sensor. We present a stochastic optimization model and use a scenario based approach to sample the location for every sensor deployed. An optimal deployment policy is determined using a heuristic solution procedure, and the expected coverage of the resulting solution is compared to a random distribution of sensors throughout the entire network. The impact of sensor failures on the model formulation is also addressed, with computational results further illustrating the benefit of the optimization model over a random deployment strategy.

Future research might focus on the impact of multiple sensor failure distributions on the resulting model formulation and methodology. As discussed in Chapter 5, the stochastic optimization model is capable of addressing heterogeneous sensor failures with minor changes to the model and scenario construction, but the remaining models require additional attention to properly reflect such a change. Chapter 4 may also consider a value function approximation based on a parametric model and basis functions, with a comparison of the resulting policy to the current lookup table approach. While the stochastic optimization model in Chapter 5 focuses on a coverage objective, the correlation between network coverage and reliability implies the solution may also be informative to the methodology in Chapters 3 and 4. Future work related to Chapter 5 may consider improving the heuristic solution performance, or efficient methods to estimate the quality of the heuristic solution compared to the coverage of an optimal deployment policy.