# RESPIRE++: Robust Indoor Sensor Placement Optimization under Distance Uncertainty

Onat Gungor, Tajana S. Rosing, *Fellow, IEEE*, and Baris Aksanli, *Member, IEEE*

Abstract—**Sensor placement in wireless sensor networks (WSN) aims to maximize coverage while minimizing total deployment cost. However, existing coverage-only approaches do not consider the robustness of the entire system where sensors may break down or malfunction. In this paper, we first propose a robustness-aware sensor placement approach by constructing a multi-objective optimization model. Our experiments demonstrate that this method increases the robustness of a WSN by up to 50%, with 201% higher probability of monitoring the entire environment as compared to the state-of-the-art coverage-only approach. The paper further improves the proposed method by introducing a robust optimization based sensor placement approach which considers the distance uncertainty between a sensor and a target. We show that this improved model increases the probability of target detection by up to 77% compared to state-of-the-art coverage-only approach.**

Index Terms—**indoor sensor placement, mathematical optimization, robust optimization, wireless sensor networks**
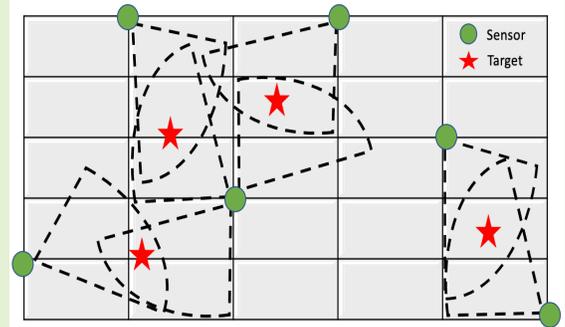
I. INTRODUCTION

SENSORS are used by many applications to monitor an environment and receive the most up-to-date information about it. Usually, this is achieved by placing a number of sensors, forming a Wireless Sensor Network (WSN) in a specific area, by considering coverage and connectivity. Sensor placement directly impacts the efficiency of the allocated resources and system performance [1]. For the best performance, applications should observe and monitor the greatest total relevant area possible. This can be achieved by a large number of sensors, but the cost of such a solution grows with more sensors. Some studies formulate an optimization problem with the goal of maximizing sensor coverage with minimum total cost, where the cost is the function of the total number of sensors [2], energy consumption [3], or communication bandwidth [4].

Sensors are usually small and fragile devices, and thus susceptible to breaking down or malfunctioning. They also

Onat Gungor is with the Electrical and Computer Engineering Department, University of California San Diego and San Diego State University, La Jolla, CA 92093 USA (e-mail: ogungor@ucsd.edu).

Tajana Rosing is with the Computer Science and Engineering Department, University of California San Diego, La Jolla, CA 92093 USA (e-mail: tajana@ucsd.edu).

Baris Aksanli is with the Electrical and Computer Engineering Department, San Diego State University, San Diego, CA 92182 USA (e-mail: baksanli@sdsu.edu).

face the physical environment and their readings can become unstable or inaccurate. The overall WSN performance can be affected badly by a possible sensor break-down or an uncertain sensor reading. Optimizing sensor placement only from a coverage perspective can cause a very sparse sensor placement, where most areas are covered by a single sensor. Here, when a sensor goes down, there is no alternative to take its place to cover the same range, decreasing the robustness of the system. Figure 1 creates a scenario to observe this robustness issue. The four sub-figures monitor an area, shown as 2×2 grid. The small circles are sensors that can detect up to 1 unit, and the middle point (represented by a face) is the point of interest (PoI) to be covered. The top-left figure shows a coverage-only approach where the PoI can be detected only by the sensor at (1,2). If this sensor breaks down (top-right), the PoI becomes undetectable. The bottom-left figure is an alternative method (robust) where the PoI can be detected by all the sensors. If the same sensor breaks down, the PoI stays detectable by two other sensors (bottom-right).

Our goal is to apply robustness to the sensor placement problem. To achieve this, we adopt the setup from [5], where we place sensors to detect PoIs in a closed room. We represent the room in 3-D, and divide it into a grid structure. A sensor can be placed at a grid point and all PoIs should be detectable. We first formulate a problem to maximize the number of detected PoIs. Here, a sensor can take only binary decisions (i.e. detect/not detect), which is not a realistic assumption. A probabilistic model is required to find the detection probability of each PoI when sensors placed in specific locations. We extend the simple model where detection depends on the distance between sensor(s) and the potential target(s). This
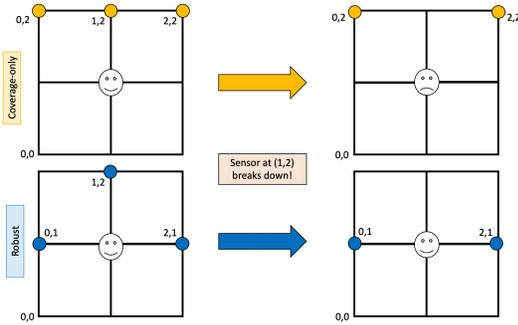
Fig. 1: Coverage-only vs. robustness-aware approaches

formulation represents the coverage-only approach from the state of the art [6], [7], our main comparison basis.

Our first contribution is formulating a robustness-aware sensor placement problem (*RESPIRE*). To quantify robustness, we define a metric called "detectability degree" for each PoI, that measures with how much probability a PoI is covered by the deployed sensors. We reformulate the coverage-only problem with respect to the detectability degrees. We run experiments with multiple room configurations to compare our approach with the coverage-only method. We show that the robustness can be increased by up to 50% compared to the coverage-only approach. We verify this robustness improvement by analyzing system coverage with broken or malfunctioning sensors. *RESPIRE* can reach up to 201% higher probability of monitoring the entire environment, compared to the coverage-only approach. This method leads to a more robust sensor placement, ensures that the application can continue to perform effectively even when there are non-functional sensors.

State-of-the-art studies and our first formulation (*RESPIRE*) both assume that the PoIs are static. This assumption might not hold in environments where PoIs are expected to move. The movement of dynamic PoIs can essentially alter the environment, and worsen the sensor network performance constructed with respect to static PoIs. We enhance our robustness-aware model by proposing a robust optimization (RO) extension for sensor placement (*RESPIRE*++). This improved model guarantees that for any realization of the distance between a sensor and a PoI, our sensor placement solution stays feasible, i.e. the sensor can still detect the PoI. We compare the performance of *RESPIRE*++ against both the coverage-only approach and our own *RESPIRE* in terms of detection capability for different PoIs placed across the test environment. We observe that for larger room sizes, *RESPIRE*++ provides a better solution. Overall, *RESPIRE*++ performs better than the coverage-only method by up to 77%.

II. RELATED WORK

Sensor placement has been studied extensively in various problem setups where coverage and connectivity are two important performance metrics [8]. While connectivity ensures reliable information transmission among sensors, coverage reflects how well a sensor field is monitored [9].

Coverage problems can be classified into three main categories [10]: point (target) coverage, area coverage and barrier coverage. Our focus is on target coverage since we have a set of targets to be detected by the deployed sensors in an indoor space. There are three main approaches to solve target coverage problems [9]. Exhaustive search enumerates all possible sensor placement solutions and chooses the best one [11]. As it has exponential complexity, it can provide solutions for only very small instances. Optimization-based approaches construct integer programming models which can be solved by conventional solvers [12]–[14]. The drawback of this method is that for larger instances, we may not obtain a solution in polynomial time. Lastly, approximation algorithms, i.e. heuristics, try to find nearly optimal solution(s) with reasonable execution time [15]. Many heuristics are developed to solve sensor placement problems [16]–[18].

Indoor sensor placement is another area where the goal is to place sensors optimally in a closed space. Vlasenko et al. [12] propose a method to deploy motion sensors to detect individual activities through mobility modeling. Feng et al. [19] present a multi-objective, mixed-integer-discrete-continuous optimization problem, and solve it by using a divide and conquer-based genetic algorithm for the optimal placement of binary sensors. The goal of the work presented by Fanti et al. [7] is to deploy sensors in a home environment. They propose three different integer linear programming (ILP) models based on sensor detection radius, angle, and orientation. In their robust model, they minimize total deployment cost and maximize overlapping detection areas simultaneously. This model can be seen as similar to ours. However, this work does not quantify overall system robustness or present experimental results verifying the system robustness.

In any real-life optimization problem, uncertainty in input data may lead to infeasible solution(s). Robust optimization (RO) is one approach to stay immune against possible changes in data. In RO, uncertain data is assumed to be in an uncertainty set where the solution should stay feasible with any sample from the uncertainty set. For sensor placement, RO is mainly used in water distribution systems. Watson et al. [20] propose robust sensor placement in contamination warning systems. Sela and Amin [14] suggest robust sensor placement in intelligent water systems with robust mixed integer optimization and robust greedy approximation approaches. Liu et al. [21] propose a more generalized robust sensor placement approach for large-scale linear time-invariant systems, based on one sensor node failure and one link failure as set cover problems. Although these works have a robust component, they did not explicitly construct and solve a robust optimization problem. In our work, we consider possible target movements in a closed space. We model these potential movements with uncertainty sets, and use them in our robust optimization based sensor placement model, *RESPIRE*++.

III. MATHEMATICAL FORMULATION

We represent the target space, that will be covered by a number of sensors as a 3-D grid. Sensors can be placed at these grid points and all the grid points (i.e. points of interest - PoIs) in the system need to be covered. Let the room consist of n_x, n_y, n_z PoIs in the *x*, *y* and *z* dimensions, respectively.

The sensor sensing range r indicates the maximum distance where a sensor is able to cover. A sensor at location (x_1, y_1, z_1) covers a PoI at (x_2, y_2, z_2) if and only if r is greater than or equal to the *Euclidean distance* between the sensor and the PoI, i.e. $r \geq \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$.

A. Coverage-only Model

We start with an initial model based on maximal covering location problem (*MCLP*). In *MCLP*, given facilities (e.g. warehouse, depot), possible facility locations, and demand locations (i.e. customers), the goal is to locate the facilities to maximize the number of demand points covered [22]. To adapt our problem to *MCLP*, we use the following parameters:

$\mathcal{N}$ = number of sensors to be located
$\mathcal{G}$ = set of PoIs to be detected
$\mathcal{S}$ = set of potential sensor locations
g = index of PoI $g \in \mathcal{G}$
s = index of possible sensor location $s \in \mathcal{S}$
r = sensor sensing range
d_{sg} = Euclidean distance between sensor and PoI

$$\xi_{sg} = \begin{cases} 1, & \text{if } d_{sg} \leq r \\ 0, & \text{otherwise} \end{cases}$$

The decision variables are as follows:

$$X_s = \begin{cases} 1, & \text{if sensor is positioned at location s} \\ 0, & \text{otherwise} \end{cases}$$

$$Y_g = \begin{cases} 1, & \text{if PoI g is detected} \\ 0, & \text{otherwise} \end{cases}$$

The integer linear programming (ILP) model becomes:

$$\text{maximize} \quad \sum_{g \in \mathcal{G}} Y_g \tag{1}$$

$$\text{subject to} \quad \sum_{s \in \mathcal{S}} \xi_{sg} X_s \geq Y_g \quad \forall g \in \mathcal{G} \tag{2}$$

$$\sum_{s \in \mathcal{S}} X_s = \mathcal{N} \tag{3}$$

$$X_s = \{0, 1\} \quad \forall s \in \mathcal{S} \tag{4}$$

$$Y_g = \{0, 1\} \quad \forall g \in \mathcal{G} \tag{5}$$

(1) is our objective function which maximizes the number of PoIs covered. Constraints (2) enable a PoI g to be covered if and only if one or more sensors are able to detect g. Constraint (3) forces to place exactly $\mathcal{N}$ sensors. Constraints (4) and (5) are binary variable constraints.

B. Coverage-only Probabilistic Model

The coverage-only model assumes that a sensor can only make binary detection decisions (i.e. detect or not). In reality, there is an uncertainty associated with sensor readings. Thus, sensor detection should be based on a probabilistic model [23]. The probability of sensor detection is related to the distance between a sensor and a PoI. To achieve this, we define p_{sg} as the detection probability of a PoI g by a sensor located

at point s. We adopt a commonly used exponential detection probability function as in Dhillon et al. [24] to demonstrate the relationship between d_{sg} and p_{sg}:

$$p_{sg} = e^{-\alpha d_{sg}} \tag{6}$$

where $\alpha \in [0, 1]$ denotes the rate at which sensor's detection probability decreases with distance. The bigger the value of α, the quicker p_{sg} will decrease as distance increases. We calculate p_{sg} values for all possible sensor-PoI tuples using Equation 6. We denote the probability of missing a PoI g with a sensor located at s as $(1 - p_{sg})$. We denote $\tau_g \in [0, 1]$ as the maximum allowable miss probability for each PoI. Larger values of τ_g lead to full coverage of the system using fewer number of sensors (flexible system), whereas smaller values of τ_g require larger number of sensors to obtain full coverage (strict system). Ultimately, we reformulate constraints (2) as:

$$\sum_{s \in \mathcal{S}} \eta_{sg} X_s \geq \zeta_g Y_g \quad \forall g \in \mathcal{G} \tag{7}$$

where $\eta_{sg} = -\ln(1 - p_{sg})$ and $\zeta_g = -\ln(\tau_g)$. In this model, the semantic meaning of Y_g also changes from the previous section. Here, it measures the number of PoIs that can satisfy constraints (7). For the rest of the paper, we call this optimization model *Coverage*, representing the state of the art.

C. Robustness-Aware Probabilistic Model (RESPIRE)

In a system with several sensors, we cannot expect every sensor to correctly and accurately function indefinitely. There might be some environmental disruptions which affect the working condition of a sensor. This might lead to sensor malfunctioning, inaccurate readings, or a complete breaking down. To exemplify, let us consider a system setup, with some sensors placed to detect all PoIs, and one PoI is covered by (binary detection) only one sensor. If this sensor is broken (or malfunctioning), then this particular PoI can no longer be covered, which threatens the correct functioning of the entire system. To prevent this, we need to place sensors in a way to increase the resilience of the system.

We construct our robust sensor placement model based on probabilistic sensor detection. We define "detectability degree" (δ_g) of a PoI, as the sum of detection probabilities from all deployed sensors to the respective PoI. δ_g is formulated as:

$$\delta_g = \sum_{s \in \mathcal{S}} p_{sg} X_s \quad \forall g \in \mathcal{G} \tag{8}$$

To understand this formulation better, consider Figure 1 where sensor detection is binary (p_{sg} is 1 if detected, otherwise it is 0). In the top-left figure, δ_g is 1, whereas in bottom-left it is 3 (i.e. PoI can be detected by three sensors.) In our case, instead of binary numbers (0 or 1), we use detection probabilities to find δ_g. We define the robustness of our system using average and minimum δ_g. μ denotes the average δ_g for all PoIs and ψ denote the minimum *detectability degree* among all PoIs where (12) and (13) provide mathematical formulations for these variables respectively. For a PoI, a higher detectability degree corresponds to a more robust and

resilient system. This is because a higher detectability degree for a PoI means that if some sensor(s) covering that point break down, there are alternatives to cover that particular point.

To better explain why we choose these two variables to measure the robustness of a system, we provide a simple sensor placement scenario. The scenario contains five PoIs to detect and seven sensors under binary sensor detection decisions. Assume that {2, 5, 6, 7, 0} gives us the number of sensors that can detect each PoI (i.e. first point is detected by two sensors, second point is detected by five sensors and so on). According to this set, the average detection value is 4 (20/5), while the minimum is 0. We see that although the average detection value is high and there are three PoIs with detection values higher than average, there is a PoI that cannot be detected by any sensor, reducing the overall system robustness. Thus, it is not a good idea to maximize only the average value, but we should also maximize the minimum value simultaneously to obtain a more balanced distribution of values. Accordingly, in our formulation, we maximize a weighted sum of the average and minimum detectability degrees across all PoIs.

We create a multi-objective optimization model where we use the weighted sum method [25]:

$$\text{maximize} \quad w_1\mu + w_2\psi \tag{9}$$

$$\text{subject to} \quad \sum_{s\in\mathcal{S}} \eta_{sg}X_s \geq \zeta_g \quad \forall g \in \mathcal{G} \tag{10}$$

$$\sum_{s\in\mathcal{S}} X_s = \mathcal{N} \tag{11}$$

$$\mu = \frac{\sum_{g\in\mathcal{G}}\sum_{s\in\mathcal{S}} p_{sg}X_s}{|\mathcal{G}|} \tag{12}$$

$$\psi \leq \sum_{s\in\mathcal{S}} p_{sg}X_s \quad \forall g \in \mathcal{G} \tag{13}$$

$$w_1 + w_2 = 1 \tag{14}$$

$$w_1, w_2 \geq 0 \tag{15}$$

(9) is our objective function which maximizes the weighted sum of average and minimum detectablity degrees. Constraints (10) are the missing probability constraints where $\eta_{sg} = -\ln(1-p_{sg})$ and $\zeta_g = -\ln(\tau_g)$. Constraint (11) forces to place exactly N sensors. Constraint (12) is an equality constraint for average detectability degree where $|\mathcal{G}|$ denotes cardinality of set $\mathcal{G}$. Constraint (13) is an inequality constraint to denote the minimum detectability degree. Constraint (14) forces sum of weights to be equal to 1. Constraint (15) ensures that weights are non-negative. In our experiments, we select the values of w_1 and w_2 as 0.5, i.e. we assign equal importance to the average and minimum detectability degrees. For the rest of the paper, we call this model *RESPIRE*.
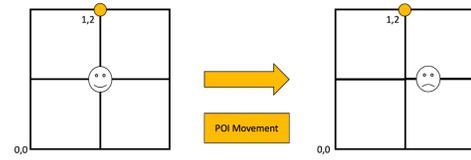

Fig. 2: Robust optimization motivating example

D. Robust Optimization Based Model (RESPIRE++)

Robust optimization (RO) is one approach to deal with the uncertainty inherent in optimization data and parameters. Uncertain data is represented by a range of values (i.e. uncertainty set) and the solution proposed by RO stays feasible for all realizations of the uncertain parameters that lie in this predetermined uncertainty set. This uncertainty set is then used to formulate the RO problem.

Previously, we defined detection probability p_{sg} depending on the distance between a sensor and a PoI (see Equation (6)). Here, each PoI is capable of moving (note that we previously assumed static PoIs). When a POI moves, the change in the distance between sensor and PoI affects the detection probability. Eventually, this change can have an impact on the constraints (10) in the *RESPIRE* model, since η_{sg} depends on p_{sg} (i.e. $\eta_{sg} = -\ln(1-p_{sg})$). In the worst-case scenario, this might even make the obtained solution infeasible. The main motivation behind the RO formulation under PoI movement uncertainty is illustrated in Figure 2. Here, there is only one sensor placed at (1,2) and one PoI at (1,1). On the left sub-figure, the sensor is able to detect the PoI. After the PoI moves to the right by a small margin, the PoI can no longer be detected since the PoI is not within the detectable range of the sensor. *RESPIRE* originally considers only static PoI locations in its formulation. We will show that under possible PoI movement, *detectability degree* (δ_g) changes.

We reformulate d_{sg} as an uncertain input variable where its value changes based on PoI movement. We know the nominal values of d_{sg}, representing the distance between a sensor and a PoI under the static PoI assumption. We need to add the largest possible magnitude of PoI movement dispersion to create our RO model. Accordingly, d_{sg} is rewritten as:

$$d_{sg} = d_{sg}^0 + \hat{d_{sg}}\gamma sg \quad \gamma sg \in [-1,1] \quad \forall s \in \mathcal{S}, \forall g \in \mathcal{G}$$

where d_{sg}^0 is the nominal distance between a sensor and a PoI, $\hat{d_{sg}}$ is the largest dispersion amount, and γsg is the primitive uncertain parameter that resides in [-1,1]. That is, $\gamma \in \mathcal{U}$ is the vector of primitive uncertainties, and $\mathcal{U}$ is an uncertainty set [26]. We can now rewrite the constraints (10):

$$\sum_{s\in\mathcal{S}} \ln(1 - e^{-\alpha(d_{sg}^0 + \hat{d_{sg}}\gamma sg)})X_s \leq \ln(\tau_g) \quad \gamma sg \in \mathcal{U}, \forall g \in \mathcal{G} \tag{16}$$

where we plug η_{sg}, ζ_g, and the reformulated d_{sg} into the inequality and we multiply both sides by -1 to flip the inequality sign. Note that this constraint becomes intractable due to its semi-infinite structure. To obtain a tractable formulation, we derive the robust counterpart using a box uncertainty set. In the box uncertainty set, $\mathcal{U}$ is selected to be $\|\gamma\|_\infty \leq 1$. The box uncertainty set contains the full range of realizations

for each component of the uncertain parameter [27]. One can think this uncertainty set as the most robust choice due to the fact that all parameters take their worst-case values at the same time. Without loss of generality, we may assume the vector of uncertain parameters is in the unit box. Accordingly, the robust counterpart of (16) using the box uncertainty set is formulated as:

$$\sum_{s \in \mathcal{S}} \ln(1 - e^{-\alpha(d_{sg}^0 + \hat{d_{sg}}\gamma sg)}) X_s \leq \ln(\tau_g)$$

$$\gamma sg \in [-1, 1]^{\mathcal{S} \times \mathcal{G}}, \forall g \in \mathcal{G}$$

(17)

It is trivial to see that the worst-case realization of the uncertain parameter that maximizes the LHS of (17) is $\gamma_{sg}^* = 1$. Hence (17) can be reformulated as:

$$\sum_{s \in \mathcal{S}} \ln(1 - e^{-\alpha(d_{sg}^0 + \hat{d_{sg}})}) X_s \leq \ln(\tau_g) \quad \forall g \in \mathcal{G} \quad (18)$$

which yields the final robust version of constraints (10). Here, we use nominal d_{sg}^0 values, and we select $\hat{d_{sg}}$ as 1.5 (PoI can move from one grid point to another in the worst case), and α as 0.576 (refer to Section IV-B for the selection of α value). For the rest of the paper, we call this robust optimization based model *RESPIRE++*.

IV. EXPERIMENTAL EVALUATION

A. 3-D Room Representation

We represent the target environment as a 3D room with height (z) and the base area (x and y). We divide the room into grids where each grid point can be a potential sensor location and all grid points should be covered by a sensor (i.e. PoIs). Notably, some points in a room are not feasible options for sensor placement, e.g. a middle grid point is infeasible since the sensor cannot hang in the air. We exclude such grid points from the set of possible sensor locations. Although we do not place sensors on the ground due to safety concerns, we allow ground sensor placement over the bottom edges of the room to enable full coverage with fewer number of sensors.

B. Experimental Setup

We implement ILP models in YALMIP [28] using the Gurobi Solver [29]. All presented optimization models (*Coverage*, *RESPIRE*, and *RESPIRE++*) find solutions offline, thus there is no real-time delay, and their offline execution time is similar. We run experiments on a PC with 16 GB RAM and an 8-core 2.3 GHz Intel Core i9 processor. For our sensor-based application, we adopt the setup from [5] with low-resolution thermal sensors. We consider different room configurations with a fixed height of $3m$ since the minimum room height should be $2.75m$, as stated in [30]. The distance between each grid point in each axis is $1.5m$. We use the following room configurations:

- small room: $4.5m \times 4.5m \times 3m$
- medium room: $6m \times 6m \times 3m$
- large room: $7.5m \times 7.5m \times 3m$
- very large room: $9m \times 9m \times 3m$
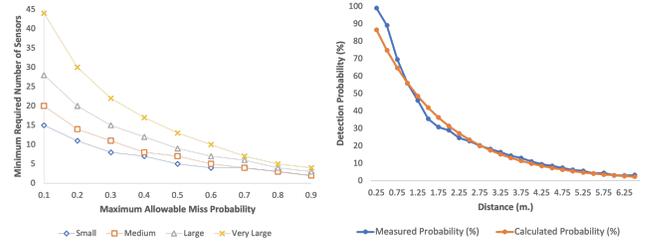
(a) Maximum allowable miss probability (τ_g) selection

(b) Best fit function for optimal α selection

Fig. 3: Experimental analysis for parameter selection

In order to determine maximum allowable miss probability (τ_g from section III.B) values, we perform an experiment where we find the minimum number of sensors for a feasible solution (i.e. a solution which can satisfy the system's coverage requirement) with varying τ_g values. Figure 3a illustrates the results of this analysis where x-axis represents different $\tau_g \in [0.1, 0.9]$, and y-axis provides the minimum number of sensors for a feasible solution. We observe that the smaller the τ_g is, the bigger the number of sensors needed. Accordingly, we select different τ_g values for each room setting. For the small room, we use 0.4 as τ_g and increment it by 0.05 for each larger setting (e.g. 0.45 as τ_g for the medium room, etc.). The selected τ_g values provide a balanced sensor placement, not too strict or flexible.

For the optimal value of α (the rate at which the sensor's detection probability decreases from Section III.B), we perform an experiment, using a thermal sensor as in [5]. In this experiment, we measure the probability of detecting the presence of a PoI with respect to increasing distance. Using the measured probability values, we use curve-fitting and obtain the optimal α as 0.576. Figure 3b shows the measured (experiment results) vs. calculated (curve-fitting results) values.

C. Results

RESPIRE: For both Coverage (state of the art) [6], [7] and RESPIRE formulations, a feasible solution means that the number of sensors and their locations can satisfy the system's coverage requirement. This is represented by Equation 10, i.e. for each PoI, the probability of non-detection should be less than the maximum allowable miss probability, τ_g. The minimum number of sensors to obtain the first feasible solution are 7, 8, 9, and 11 sensors for small, medium, large, and very large rooms respectively. The left hand side (LHS) of Equation 10 for each PoI indicates how well the point is covered. As the LHS for each PoI gets bigger, the point is covered with higher probability, which means that the PoI is less prone to sensor break downs.

To understand how robust a sensor placement is, we create a scenario where one of the initially-placed sensor breaks down. Then, we calculate the LHS values of Equation 10 for each PoI for both *Coverage* and *RESPIRE*. We calculate the minimum and average LHS values across all points, excluding the broken sensor. The minimum value shows the most vulnerable point, while the average measures the vulnerability
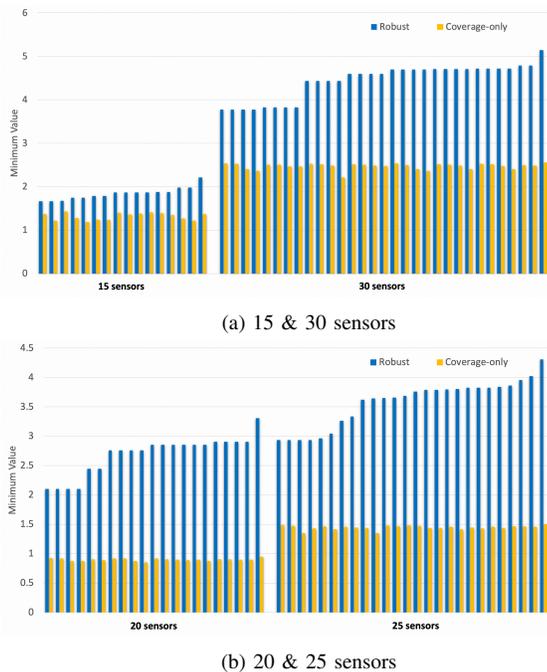
(a) 15 & 30 sensors

(b) 20 & 25 sensors

Fig. 4: Small room minimum detectability degree: *Coverage* [6], [7] vs. *RESPIRE* (a) 15-30 sensors (b) 20-25 sensors

TABLE I: Minimum value improvement with 1 broken sensor

# Sensors	Small	Medium	Large	Very Large
15	38%	67%	33%	26%
20	196%	146%	101%	82%
25	146%	180%	126%	108%
30	79%	201%	131%	94%

TABLE II: Average value improvement with 1 broken sensor

# Sensors	Small	Medium	Large	Very Large
15	2.7%	5.4%	5.9%	7.8%
20	0.9%	0.4%	3.7%	7.2%
25	0.5%	0.8%	1.7%	4.9%
30	-0.2%	0.8%	0.7%	7.4%

TABLE III: Minimum value improvement with 2 broken sensors

# Sensors	Small	Medium	Large	Very Large
15	36%	64%	35%	33%
20	176%	131%	99%	89%
25	131%	158%	119%	102%
30	70%	201%	127%	93%

across all points. We first show this analysis in detail for the small room in Figure 4. The figure includes 4 cases, with 15 and 30 sensors in a); 20 and 25 sensors in b). For each case, we calculate the minimum LHS value across all points when a particular sensor breaks down, for both *Coverage* and *RESPIRE* models. The x-axis indicates different broken sensor cases, i.e. each blue (*RESPIRE*) / yellow (*Coverage*) column pair represents a particular broken sensor. For each case, the right-most two columns represent no broken sensor case. We see that across all sensor configurations, *RESPIRE* leads to significantly higher minimum LHS values when a sensor breaks down, i.e. the most vulnerable point with our model has a much higher probability of detection as compared to the *Coverage* case. This observation signifies that *RESPIRE* sensor placement is less affected by broken sensors.

We expand this analysis on all room settings with 15, 20, 25 and 30 sensors, comparing the minimum (Table I) and average (Table II) LHS value improvement of *RESPIRE* against *Coverage*. We see that the average values change up to 7.8%, whereas the minimum value change is up to 201%. The minimum value change is more important as the minimum value shows the most vulnerable point. We see that *RESPIRE* sensor placement makes the most vulnerable point significantly less prone to broken sensors, increasing the robustness of the overall system.

We further conduct experiments for two broken sensor cases where we consider all possible sensor tuples as broken for all room settings. We observe comparable minimum value improvement values as shown in Table III with respect to the one sensor case. We obtain very similar average values results at almost all instances (thus, not included for clarity).

Next, we use Equation 9, the weighted sum of average and minimum detectability degrees, as a metric to quantify the robustness of a sensor placement. In the broken sensor analyses above, we observe that the *Coverage* model might obtain high coverage levels, but it neglects the average and minimum detectability degrees, leading to a more vulnerable and less robust system. We calculate Equation 9 across all room settings with different number of sensors, shown in Figure 5. For the small room, the biggest improvement is with 20 sensors whereas in the medium room it is with 30 sensors which is the maximum improvement (50%) among all room and sensor configurations. Table IV provides the improvements based on the robustness metric (in terms of percentage average and standard deviations) for all types of room configurations. We observe that the biggest improvement is with the medium room, which aligns with the broken sensor analyses, where the highest minimum LHS difference is with the medium room again (Table I and III).

RESPIRE++: Due to possible movements of PoIs, the proposed solution by *Coverage* or *RESPIRE* may become inefficient or even infeasible. In *RESPIRE++*, we ensure that the proposed solution stays immune against the possible changes in the distance between sensors and PoIs. To demonstrate the advantage of *RESPIRE++*, we create an experimental scenario where PoIs are placed in a room independently, based on a Normal distribution. Thus, they can be placed not only at grid points but anywhere in between as well. By using the sensor placement solutions proposed by *Coverage*, *RESPIRE*, and *RESPIRE++*, we calculate the detectability degree (δ_g) for each PoI.

We first create 10 independent 3-D PoIs (x, y, and z coordinates) to be distributed across different rooms. Random variables for x and y coordinates are generated based on the following Normal distributions $\mathcal{N}(\mu, \sigma^2)$:

- small room $\sim \mathcal{N}(2.25, 0.75^2)$
- medium room: $\sim \mathcal{N}(3, 1^2)$
- large room: $\sim \mathcal{N}(3.75, 1.25^2)$
- very large room: $\sim \mathcal{G}(4.5, 1.5^2)$

For the z coordinate, we use the same distribution $\mathcal{N}(1.5, 0.5^2)$ across all room types. The reason behind the Normal distribution and its parameter selection is intuitive.
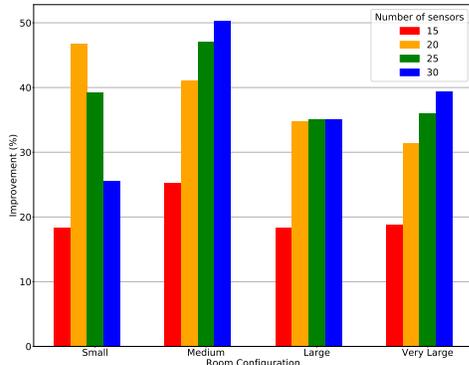
Fig. 5: Robustness improvement vs. number of sensors

TABLE IV: Average robustness improvement

Room Configuration	Robustness Improvement (%)
Small	32±13
Medium	41±11
Large	31±8
Very Large	31±9

(a) **Sheet metal machinery**   (b) **Sheet metal shop floor plan**

Fig. 6: Sheet metal shop floor [31]

Fig. 7: Sheet metal shop floor layout [31]

The probability that a PoI is in the center of the room is more likely than the edges (where this probability is maximum at the center of the room). For the parameters, we use the 3σ rule to cover all possible PoIs. For instance, in the small room, we allow the random variable to take values from 0 to 4.5. We then solve three different optimization problems: *Coverage*, *RESPIRE*, and *RESPIRE++* under different sensor placement scenarios. Given the solutions by these three models, we measure the detectability degree δ_g for each PoI. We then calculate the average δ_g values across all PoIs. The higher the δ_g value is, the better the model is.

Table V presents the average δ_g values after 10 PoIs are distributed across all room types with 30 sensors placed. In small and medium rooms, *RESPIRE++* performance is worse than *Coverage*. This is because PoI movement is limited in those rooms (i.e. there are fewer places to move to due to the small room size). However, as the room size gets bigger, *RESPIRE++* starts to surpass other methods. Compared to *RESPIRE*, for the large room, we obtain 1% average (up to 6%) δ_g improvement; and for the very large room, we have 6% average (up to 32%) δ_g improvement. Against *Coverage*, *RESPIRE++* provides up to 2% improvement on average (up to 10%) in the large room. For the very large room, *RESPIRE++* is better than *Coverage* by 15% on average (up to 34%).

TABLE V: Average δ_g values for 30 sensor placement

	Small	Medium	Large	Very Large
Coverage	5.67	3.84	2.64	1.81
RESPIRE	5.66	3.78	2.68	1.97
RESPIRE++	5.64	3.79	2.71	2.08

Sheet Metal Shop Floor Sensor Placement: To show the generalizability of *RESPIRE++*, we create an Industrial Internet of Things (I-IoT)-oriented case study, where the goal is to place sensors optimally in a sheet metal shop floor. The shop floor is defined as the place in a manufacturing facility where assembly or production is performed. For the sheet metal shop floor, we may have different machinery and equipment as illustrated in Figure 6. We use the sheet metal shop layout suggested by [31]. This layout is shown in Figure 7. Accordingly, we consider a room in the same size of this metal shop ($15m \times 7.5m$) which has a height value of $7.5m$ as stated in [32]. For this experiment, we place 100 sensors using *Coverage*, *RESPIRE*, and *RESPIRE++*. Similar to previous experiments, we place 10 PoIs across the room based on Normal distribution. For the x coordinate, PoI location $\sim \mathcal{N}(7.5, 2.5)$, and for the y, and z coordinates, PoI location $\sim \mathcal{N}(3.75, 1.25)$. Note that each PoI is sampled independently from each other. In this setup, one can think of the PoIs as the raw material, work-in-progress, or finished products. Since these PoIs have a highly dynamic structure (i.e. need to transfer from one station to another), RESPIRE++ can be more beneficial than other approaches. Figure 8 demonstrates the individual PoI detectability degrees for the three different approaches which are represented with different colors. In this figure, x-axis represents the PoI number, and the y-axis has the normalized detectability degree values. We normalize δ_g values across each PoI independently. We observe that *RESPIRE++* is the best approach at all PoI instances. On average, *RESPIRE++* improves *RESPIRE* performance by 4% (up to 17%). Compared to *Coverage*, we obtain 48% average (up to 77%) improvement.

Lastly, we repeat the broken sensor experiment for the sheet metal shop floor setup (using the same experimental setup) to further showcase the robustness of *RESPIRE++*. Similar to the previous experiments, we assume that one of the initially placed sensors breaks down. Then, we calculate the LHS values of Equation 10 and measure the minimum value across all PoIs. Note that the minimum LHS value across all PoIs shows how detectable the most vulnerable point is. The bigger this value is, the more detectable a PoI is. Here, we observe that
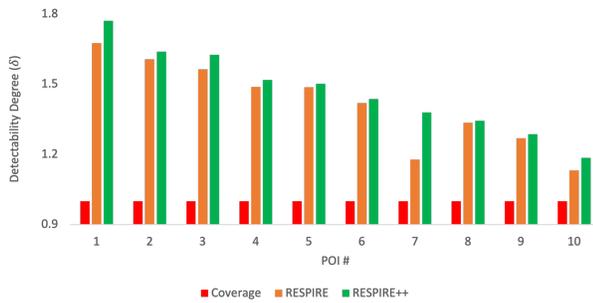
Fig. 8: Detectability degree comparison for sheet metal shop floor sensor placement

RESPIRE++ provides the largest minimum values under all possible sensor break-down scenarios. Specifically, *Coverage*, *RESPIRE*, and *RESPIRE++* has 0.45, 0.65, and 0.68 minimum values respectively. This means that *RESPIRE++* improves the robustness of a system by 51.1% over *Coverage*, and by 4.6% over *RESPIRE*. These results show that *RESPIRE++* makes the most vulnerable point significantly less prone to broken sensors which improves the robustness of the system compared to both *Coverage* and *RESPIRE*.

V. Conclusion

Sensors are prone to breaking down and imprecise readings, thus the performance of a sensor-based application can be heavily impacted by missing sensors. This paper proposes two novel sensor placement methods, that can maintain the coverage of a sensor field even when sensors break down or malfunction. In our first method, we construct a robustness-aware, probabilistic, multi-objective integer linear programming model which maximizes the weighted sum of minimum and average detectability degrees of targets. This method increases the robustness of a sensor-based system by up to 50% compared to the state-of-the-art coverage-only approach. It also has up to 201% higher probability of monitoring the entire environment, compared to state-of-the-art coverage-only approach. We then improve this model to consider the distance uncertainty between targets and sensors. To address this uncertainty, we propose a robust optimization approach which is guaranteed to stay feasible under any target movement. We show that this method increases the detection ability of sensors by up to 77% compared to the state-of-the-art sensor placement approach.

References

[1] A. T. Murray, K. Kim, J. W. Davis, R. Machiraju, and R. Parent, "Coverage optimization to support security monitoring," *Computers, Environment and Urban Systems*, vol. 31, no. 2, pp. 133–147, 2007.

[2] O. Moh'd Alia and A. Al-Ajouri, "Maximizing wireless sensor network coverage with minimum cost using harmony search algorithm," *IEEE Sensors Journal*, vol. 17, no. 3, pp. 882–896, 2016.

[3] C. Yang and K.-W. Chin, "On nodes placement in energy harvesting wireless sensor networks for coverage and connectivity," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 27–36, 2016.

[4] Y. Pei and M. W. Mutka, "Joint bandwidth-aware relay placement and routing in heterogeneous wireless networks," in *2011 IEEE 17th International Conference on Parallel and Distributed Systems*, pp. 420–427, IEEE, 2011.

[5] S. Shelke and B. Aksanli, "Static and dynamic activity detection with ambient sensors in smart spaces," *Sensors*, vol. 19, no. 4, p. 804, 2019.

[6] İ. K. Altınel, N. Aras, E. Güney, and C. Ersoy, "Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks," *Computer Networks*, vol. 52, no. 12, pp. 2419–2431, 2008.

[7] M. P. Fanti, M. Roccotelli, G. Faraut, and J.-J. Lesage, "Smart placement of motion sensors in a home environment," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 894–899, IEEE, 2017.

[8] H. Xu, J. Zhu, and B. Wang, "On the deployment of a connected sensor network for confident information coverage," *Sensors*, vol. 15, no. 5, pp. 11277–11294, 2015.

[9] B. Wang, "Coverage problems in sensor networks: A survey," *ACM Computing Surveys (CSUR)*, vol. 43, no. 4, pp. 1–53, 2011.

[10] A. Maheshwari and N. Chand, "A survey on wireless sensor networks coverage problems," in *Proceedings of 2nd International Conference on Communication, Computing and Networking*, pp. 153–164, Springer, 2019.

[11] F. Y. Lin and P.-L. Chiu, "A near-optimal sensor placement algorithm to achieve complete coverage-discrimination in sensor networks," *IEEE Communications Letters*, vol. 9, no. 1, pp. 43–45, 2005.

[12] I. Vlasenko, I. Nikolaidis, and E. Stroulia, "The smart-condo: Optimizing sensor placement for indoor localization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 3, pp. 436–453, 2014.

[13] M. P. Fanti, G. Faraut, J.-J. Lesage, and M. Roccotelli, "An integrated framework for binary sensor placement and inhabitants location tracking," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 1, pp. 154–160, 2016.

[14] L. Sela and S. Amin, "Robust sensor placement for pipeline monitoring: Mixed integer and greedy optimization," *Advanced Engineering Informatics*, vol. 36, pp. 55–63, 2018.

[15] A. N. Njoya, C. Thron, J. Barry, W. Abdou, E. Tonye, N. S. L. Konje, and A. Dipanda, "Efficient scalable sensor node placement algorithm for fixed target coverage applications of wireless sensor networks," *IET Wireless Sensor Systems*, vol. 7, no. 2, pp. 44–54, 2017.

[16] Y. Chae and D. N. Wilke, "Heuristic linear algebraic rank-variance formulation and solution approach for efficient sensor placement," *Engineering Structures*, vol. 153, pp. 717–731, 2017.

[17] J. A. Grant, A. Boukouvalas, R.-R. Griffiths, D. S. Leslie, S. Vakili, and E. M. De Cote, "Adaptive sensor placement for continuous spaces," *arXiv preprint arXiv:1905.06821*, 2019.

[18] R. Hou, Y. Xia, Q. Xia, and X. Zhou, "Genetic algorithm based optimal sensor placement for l1-regularized damage detection," *Structural Control and Health Monitoring*, vol. 26, no. 1, p. e2274, 2019.

[19] G. Feng, Y. Yang, X. Guo, and G. Wang, "Optimal design of infrared motion sensing system using divide-and-conquer based genetic algorithm," in *2013 IEEE International Conference on Mechatronics and Automation*, pp. 482–487, IEEE, 2013.

[20] J.-P. Watson, R. Murray, and W. E. Hart, "Formulation and optimization of robust sensor placement problems for drinking water contamination warning systems," *Journal of Infrastructure Systems*, vol. 15, no. 4, pp. 330–339, 2009.

[21] X. Liu, S. Pequito, S. Kar, Y. Mo, B. Sinopoli, and A. Aguiar, "Minimum robust sensor placement for large scale linear time-invariant systems: a structured systems approach," in *4th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys)*, pp. 417–424, 2013.

[22] O. Karasakal and E. K. Karasakal, "A maximal covering location model in the presence of partial coverage," *Computers & Operations Research*, vol. 31, no. 9, pp. 1515–1526, 2004.

[23] R. R. Brooks and S. S. Iyengar, *Multi-sensor fusion: fundamentals and applications with software*. Prentice-Hall, Inc., 1998.

[24] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections," in *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.(IEEE Cat. No. 02EX5997)*, vol. 2, pp. 1581–1587, IEEE, 2002.

[25] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and multidisciplinary optimization*, vol. 41, no. 6, pp. 853–862, 2010.

[26] B. Balcik and İ. Yanıkoğlu, "A robust optimization approach for humanitarian needs assessment planning under travel time uncertainty," *European Journal of Operational Research*, vol. 282, no. 1, pp. 40–57, 2020.

[27] B. L. Gorissen, İ. Yanıkoğlu, and D. den Hertog, "A practical guide to robust optimization," *Omega*, vol. 53, pp. 124–137, 2015.

[28] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *2004 ICRA*, pp. 284–289, IEEE, 2004.

[29] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2021.

[30] M. I. Hamakareem, "Minimum height and size standards for rooms in buildings." URL: https://theconstructor.org/building/rooms-minimum-height-size-standards/5116/, Nov 2018.

[31] SNIPS, "The importance of good sheet metal machinery layout," 2017.

[32] T. Moorse, "Clear height considerations," 2015.