CAHEROS: Constraint-Aware HEuristic Approach for RObust Sensor Placement

Onat Gungor[1,3], Tajana Rosing[2], and Baris Aksanli[3]

[1]Department of Electrical and Computer Engineering, University of California San Diego
[2]Department of Computer Science and Engineering, University of California San Diego
[3]Department of Electrical and Computer Engineering, San Diego State University
ogungor@ucsd.edu, tajana@ucsd.edu, baksanli@sdsu.edu

***Abstract*—Coverage-only sensor placement lacks two important aspects: uncertainty in sensor measurements and possible sensor break-down. To incorporate these components into a sensor placement framework, we first propose our robust sensor placement model. In order to solve this model for large instances efficiently, we adapt different heuristics where we integrate constraints into the heuristic framework. To the best of our knowledge, we are the first ones adapting constraint-aware heuristics for robust sensor placement. Experimentally, by using a factory shop floor, we show that the greedy algorithm outperforms other heuristics while requiring more computational overhead. Compared to the coverage-only approach, our robust heuristics improves system robustness by up to 48%. Our heuristics can find a solution 400x faster than conventional solvers while only being 5.5% less optimal.**

***Index Terms*—sensor placement, mathematical optimization, heuristics, wireless sensor networks, internet of things**

I. Introduction and Related Work

Where to place sensors to form a wireless sensor network is a crucial decision affecting overall system performance [1]. Coverage and connectivity are two main metrics considered in sensor placement literature [2]. While coverage deals with how well a sensor field is monitored, connectivity provides reliable information transmission among sensors [3]. However, there are two main problems related to sensor placement approaches considering only these metrics: 1) there is an uncertainty in sensor measurements and 2) sensors are prone to break down. The first problem can be solved by adapting a probabilistic sensor detection [4], and for the latter one, we need a robust sensor placement perspective [5]. Robust sensor placement aims to maximize the detection ability of Points of Interests (PoIs) so that under possible sensor failures, POIs can still be detected by a certain probability. In our work, we propose a probabilistic robust sensor placement approach by maximizing the detection ability of the overall system and the most vulnerable PoIs simultaneously.

To solve a sensor placement problem, there are 3 main approaches [3]: 1) exhaustive search enumerates all possible sensor placement solutions and chooses the best one [6], 2) optimization-based approaches construct integer programming models which can be solved by conventional solvers [7], 3) heuristics that try to find nearly optimal solution(s) with reasonable execution time [8]. We first propose integer linear programming (ILP) model for robust sensor placement. Since ILP is NP-complete, it is not possible to find a solution for large instances efficiently. To solve that, we adapt different constraint included heuristics for our problem. In the literature, many heuristics are developed to solve sensor placement problem such as linear algebraic rank-variance [9], continuum-armed bandit problem formulation [10], and genetic algorithm [11]. For instance, Senouci et al. [12] propose four different sensor placement heuristics: best Score, k-Region, max need and efficient. There are also many studies approaching the problem from geometrical perspective [13], [14], [9] where they use jigsaw-based, Delaunay triangulation based and rank variance based sensor placement algorithms respectively. However, none of these studies consider the robustness of the overall system in their sensor placement approaches. In our work, we utilize different heuristics for robust sensor placement different than the state-of-the-art.

In this paper, we first present our ILP formulation for robust sensor placement problem. In order to solve this ILP for large instances efficiently (ILPs become intractable for large instances), we propose a constraint-aware heuristic approach where constraint is included using a penalty function. We adapt four different constraint-aware heuristic methods: greedy algorithm (GA), simulated annealing (SA), whale optimization algorithm (WOA), and randomized algorithm (RA). To the best of our knowledge, this is the first time that WOA is implemented for sensor placement. None of these heuristics had been previously applied to robust sensor placement. Experimentally, we use the sheet metal shop floor layout test-bed suggested by [15] where we place different number of sensors on this shop floor. We found out that GA provides the highest robustness level while requiring quite a bit of computational power. We also implement coverage-only versions of the selected heuristics where the only goal is to maximize the number of Points of Interest (PoIs) detected. Compared to coverage-only versions of these heuristics, we obtain up to 48% (22% on average) robustness improvement at our robust greedy algorithm using the objective function of our robust optimization model. Furthermore, we analyze the benefit of using heuristics over finding an optimal solution. We show that GA can deliver a solution 400x faster while only resulting in 5.5% worse robustness level.

II. Proposed Framework

A. Robust optimization mathematical formulation

We first present our multi-objective integer linear programming (ILP) model based on our previous works [16], [17]:

$$\max_{X_s} \quad 0.5\mu + 0.5\psi \tag{1}$$

$$\text{subject to} \quad \sum_{s=1}^{\mathcal{S}} X_s = \mathcal{N} \tag{2}$$

$$\sum_{s=1}^{\mathcal{S}} \ln(1 - e^{-\alpha(d_{sg}^0 + \hat{d_{sg}})}) X_s \leq \ln(\tau_g) \quad \forall g = 1, \ldots, \mathcal{G} \tag{3}$$

$$\frac{\sum_{g=1}^{\mathcal{G}} \sum_{s=1}^{\mathcal{S}} p_{sg} X_s}{\mathcal{G}} = \mu; \quad \sum_{s=1}^{\mathcal{S}} p_{sg} X_s \geq \psi \quad \forall g = 1, \ldots, \mathcal{G} \tag{4}$$

(1) is our objective function which maximizes the weighted sum of average (μ) and minimum (ψ) detection probabilities where p_{sg} denotes the detection probability of PoI g by sensor s. Constraint (2) forces to place exactly N number of sensors where X_s denotes our binary decision variable to place sensor at location s or not. Constraints (3) are the robust missing probability constraints which ensures that each PoI is detected by a certain amount of probability for all realizations of the distance between sensor s and PoI g. Here, $\alpha \in [0, 1]$ denotes the rate at which sensor's detection probability decreases with distance, d_{sg}^0 is the nominal distance between a sensor and a PoI, $\hat{d_{sg}}$ is the largest distance dispersion amount, and τ_g denotes the maximum allowable miss probability for each PoI. Constraints (4) are for average and minimum detection probabilities where $\mathcal{G}$ denotes total number of PoIs and $\mathcal{S}$ denotes total number of sensor placement locations.

B. Constraint-handling heuristic approach

In order to solve the proposed ILP for large instances, we need an efficient solution approach. Heuristic methods find an approximate solution to the optimization problems quickly. In general, heuristics do not deal with the constraints of a given optimization problem. For coping up with the constraints, there are variety of approaches such as penalty method, separation of objectives, stochastic ranking, and etc [18]. In our work, we use the penalty method where constrained problem is converted into its unconstrained version by adding constraint to the objective function. Specifically, we combine our objective function (1) with the constraints (3). Note that the constraint (2) can be included in heuristic implementation by specifying number of sensors to be placed at each iteration. Furthermore, constraints (4) and (5) are the detectability metrics (average and minimum detection probabilities) calculated after sensors are deployed. Hence, solely including constraints (3) is sufficient for a constraint-handling heuristic approach for our problem. To demonstrate how we implement this approach, consider a simple mathematical optimization problem:

$$\max_{\boldsymbol{x}} \; f_0(\boldsymbol{x}) \quad \text{s.t.} \; f_g(\boldsymbol{x}) \leq 0, \quad \forall g = 1, \ldots, \mathcal{G} \tag{5}$$

The new objective function (penalty-induced objective) is defined as $\xi(\boldsymbol{x}) = f_0(\boldsymbol{x}) - p(\boldsymbol{x})$ where $p(\boldsymbol{x})$ is the penalty term which is defined as [18]:

$$p(\boldsymbol{x}) = \sum_{g=1}^{\mathcal{G}} \theta_g max\,(0, f_g(\boldsymbol{x}))^2 \tag{6}$$

where $\theta_g \geq 0$ is a penalty factor. For our problem, $f_g(\boldsymbol{x})$ is:

$$\sum_{s=1}^{\mathcal{S}} \ln(1 - e^{-\alpha(d_{sg}^0 + \hat{d_{sg}})}) X_s - \ln(\tau_g) \quad \forall g = 1, \ldots, \mathcal{G} \tag{7}$$

and $f_0(\boldsymbol{x})$ corresponds to our objective function (1) and θ_g is selected as 10^{14} for all PoIs based on [19]. At each iteration, we compare penalty-induced objective value $\xi(\boldsymbol{x})$ to update the best solution. The penalty-induced objective function decreases the fitness of infeasible solutions and favoring the feasible solutions simultaneously [19]. To the best of our knowledge, we are the first ones adapting constraint-handling heuristics for robust sensor placement problem.

C. Robust heuristic methods

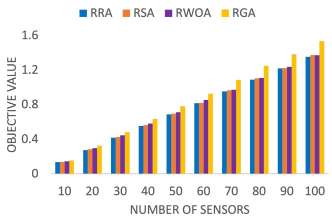
This section presents our robust heuristic approaches. Our contribution here is to add the robustness aspect to each heuristic algorithm by calculating the penalty-added objective value $\xi(\boldsymbol{x})$ and discovering the best sensor placement configuration. Best sensor placement configuration is updated at each iteration if the value of $\xi(\boldsymbol{x})$ is improved.

Robust Greedy Algorithm (RGA): GA is a well-known constructive heuristic (due to its simplicity and good performance) where optimal choice is selected independently at each time step to find the overall optimal way [20].

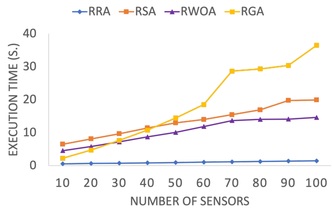
Robust Simulated Annealing (RSA): SA is a metaheuristic type algorithm which iteratively improves the possible solution(s) [21]. SA is based on a cooling process of a material. It avoids from local optima by changing the temperature dynamically and updating the best solution iteratively.

Robust Whale Optimization Algorithm (RWOA): WOA is based on the special hunting behavior of humpback whales [22]. Their prey search is divided into exploitation and exploration phases. In exploitation (bubble-net attacking method), they either shrink the circle they create to obtain smaller search space or update their spiral position around the prey. In exploration, whales randomly search for prey based on each other's positions. To the best of our knowledge, this algorithm has not been applied to the sensor placement problem. In our work, we treat prey as PoIs to be detected, and whales as the sensors to be placed.

Robust Randomized Algorithm (RRA): We also construct a baseline algorithm to compare implemented heuristic performances. Our baseline method is based on a randomized approach where at each iteration some random location is selected for sensor placement. Since this is the simplest heuristic approach one can create, we treat RA as our baseline. We expect previously selected heuristics to perform better than the randomized approach.



(a) **Performance Analysis**

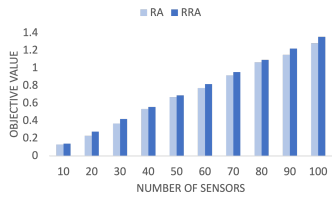


(b) **Execution Time Analysis**

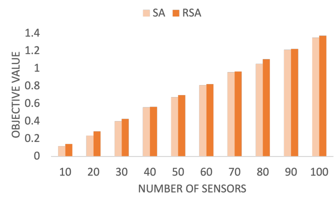
Fig. 1: Heuristic Methods Experimental Analysis

TABLE I: Robustness Improvement (%) Over Coverage-only

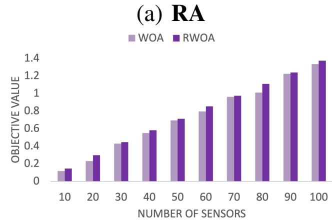
Improvement (%)	*RA*	*SA*	*WOA*	*GA*
Maximum	21.7	21.6	30.2	48.1
Average	7.5	6.4	9.1	22.1



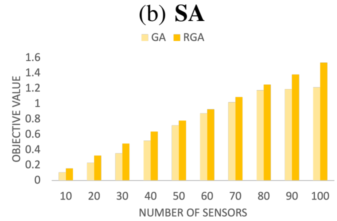
(a) **RA**



(b) **SA**



(c) **WOA**



(d) **GA**

Fig. 2: State-of-the-art Comparison

III. Experimental Analysis

We use the sheet metal shop layout from [15] as a testbed, where we model a room with the same size ($15m \times 7.5m$) and a height of $7.5m$ as in [23]. The heuristic methods have the following parameters: number of iterations=500, number of replications=10, and number of search agents=10. We run all experiments on a PC with 16 GB RAM and an 8-core 2.3 GHz Intel Core i9 processor. We select the following parameters for our robust model based on thorough experimental analysis [17]: α=0.576, d^0_{sg}=1.5, $\hat{d_{sg}}$=1.5, τ_g=0.75, $p_{sg}=e^{-\alpha d^0_{sg}}$.

Performance Analysis: We place different number of sensors ranging from 10 to 100 and calculate our objective function value (1), which measures system robustness. Higher objective value means that if some sensor(s) covering that point break down, there are alternatives to cover that particular point. Hence, the higher this value is, the more robust the system becomes. Fig. 1a presents the performances of the robust heuristics with different number of sensors. In this figure, x-axis shows the number of sensors, and y-axis has the objective function value. Robust greedy algorithm (RGA) is the best heuristic across all settings. All selected heuristics outperform robust randomized algorithm (RRA). RGA can improve the system robustness level by up to 18% (15% on average) compared to RRA. Overall, the robustness performance of the heuristics are as follows: RGA > RWOA > RSA > RRA.

Computational Overhead Analysis: Fig. 1b presents the execution time (in seconds) of the heuristics with respect to sensor placement scenarios. The more sensors we place, the longer it takes for any algorithm to complete. Specifically, GA becomes the slowest algorithm as we increase number of sensors. Note that there is a trade-off between execution time and performance. Even though GA has the highest robustness, it takes the longest to execute. It means that there is no algorithm which can provide both great robustness level and fast execution time. For instance, at 100 sensors GA is 12% better than WOA while requiring 150% more execution time.

Comparison with the Optimal Solution: When we try to solve this problem with state-of-the-art solvers that provide an optimal solution, we cannot obtain a feasible solution with up to 53 sensors. This is because all PoIs should be detected by a certain amount of probability, and using less than 53 sensors does not satisfy this condition. After passing the feasible solution threshold, it takes more than 120 minutes to find the optimal solution. Specifically, we run the optimization model for 60 sensors and observe that RGA has 5.5% less robustness compared to the optimal solution with 400x faster execution. This shows that our proposed heuristic methods are able to provide robust sensor placement solutions efficiently while not sacrificing from the optimal solution.

Comparison with Coverage-Only Heuristics: We implement coverage-only versions of the heuristics to reflect the state-of-the-art [24]. The coverage-only methods aim to maximize the number of PoIs detected [25]. Fig. 2 shows this analysis. Each sub-figure compares specific heuristic method's coverage-only and robust versions using our robustness metric (Equation 1). Our robust heuristics outperform coverage-only heuristics for all test cases. Table I presents the robustness improvement of our robust heuristics over coverage-only. We obtain up to 21.7%, 21.6%, 30.2%, and 48.1% robustness improvement for RA, SA, WOA, and GA, respectively with 7.5%, 6.4%, 9.1%, and 22.1% average improvements.

IV. Conclusion

Solely maximizing the number of PoIs detected for a sensor placement application does not deal with the possible sensor malfunctioning or probabilistic sensor detection. Hence, there is a need for a robust probabilistic sensor deployment approach. In this paper, we propose a robust mathematical model for an indoor sensor placement context. We adapt variety of heuristics to solve the optimization problem efficiently for large instances. Experimentally, we show that greedy algorithm brings the highest robustness level to the system, yet it requires high computational overhead. Furthermore, our robust heuristic methods increase the system robustness by up to 48% compared to coverage-only approach. Compared to optimal solution, our heuristic approaches can provide significantly faster solutions while keeping robustness at a certain level.

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] 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.

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

[4] V. Akbarzadeh, C. Gagné, M. Parizeau, M. Argany, and M. A. Mostafavi, "Probabilistic sensing model for sensor placement optimization based on line-of-sight coverage," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 2, pp. 293–303, 2012.

[5] C. Yang, Z. Lu, and Z. Yang, "Robust optimal sensor placement for uncertain structures with interval parameters," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 2031–2041, 2018.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] M. R. Senouci and A. Abdellaoui, "Efficient sensor placement heuristics," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2017.

[13] S.-C. Huang, H.-Y. Chang, and K.-L. Wu, "A jigsaw-based sensor placement algorithm for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 11, p. 186720, 2013.

[14] C.-H. Wu, K.-C. Lee, and Y.-C. Chung, "A delaunay triangulation based method for wireless sensor network deployment," *Computer Communications*, vol. 30, no. 14-15, pp. 2744–2752, 2007.

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

[16] O. Gungor, T. S. Rosing, and B. Aksanli, "Respire: Robust sensor placement optimization in probabilistic environments," in *2020 IEEE Sensors*, pp. 1–4, IEEE, 2020.

[17] O. Gungor, T. S. Rosing, and B. Aksanli, "Respire++: Robust indoor sensor placement optimization under distance uncertainty," *IEEE Sensors Journal*, 2021.

[18] E. Mezura-Montes and C. A. C. Coello, "Constraint-handling in nature-inspired numerical optimization: past, present and future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173–194, 2011.

[19] Q.-V. Pham, S. Mirjalili, N. Kumar, M. Alazab, and W.-J. Hwang, "Whale optimization algorithm with applications to resource allocation in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4285–4297, 2020.

[20] D. Jungnickel, "The greedy algorithm," in *Graphs, Networks and Algorithms*, pp. 135–161, Springer, 2013.

[21] E. Aarts and J. Korst, "Simulated annealing and boltzmann machines," 1988.

[22] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, pp. 51–67, 2016.

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

[24] 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.

[25] I. 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.