2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

An Integer Programming Approach for Angular Coverage under Uncertainty

Yuntian Zhang[1], Chen Chen[1], Shuxin Ding[3], and Fang Deng[1,2]

***Abstract*— This paper investigates angular coverage under uncertainty (ACU). A compact integer programming (IP) formulation is developed to model the angular field-of-view (FoV) of sensors and probabilistic coverage under uncertainty. The IP formulation minimizes the weighted non-coverage probability over the target set as well as considering the practical co-location and budget constraints. Recognizing the non-linearity, non-convexity, and non-separability of ACU, we first introduce the reformulation-linearisation technique (RLT) to obtain a tractable mixed-integer linear programming model which provides a tight lower bound for the original problem. Further, we exploit the structure of the mathematical model and customize a branch-and-cut (B&C) algorithm to solve the derived problem exactly. We show that the solution for the derived problem can also solve the original problem based on the bounding scheme. Computational experiments on a series of problem instances ranging from moderate to large size scaling up to 4,000 dimensional decision variables reveal the effectiveness and efficiency of the proposed exact approach.**

I. Introduction

Location covering problem [1], [2] is a classic and important combinatorial optimization problem. In general, the location covering problem can be divided into two categories, including maximizing coverage with a fixed cost and minimizing cost to ensure required coverage. The former leads to mathematical models such as maximal covering location problem (MCLP) [3] and its variants, while the latter comprises location set covering model (LSCP) [4] and its variants.

Among these classical formulations, there still exists some basic assumptions which should be reconsidered in real world applications:

- *The coverage is radial*. That is, a sensor covers a target if the distance metric between them is less than or equal to the coverage radius. However, in the camera-based surveillance network, the field-of-view (FoV) is angular rather than circular. This makes angular coverage an important and challenging issue in the optimization task.
- *The coverage is deterministic*. It is assumed that a target can be covered by a sensor within coverage. On the contrary, due to the uncertainty in the real world, there exists a non-zero probability that sensor coverage may not be provided to a target even within the desired coverage standard [1]. This calls for reliable probabilistic coverage under uncertainty.

There are works focusing on either angular or reliable coverage. Reference [5] determines the optimal camera placement in order to achieve the angular coverage for a given region continuously. Reference [6] formulates the angular coverage problem in two dimensions which extends the MCLP with angular covering characteristic as well as maximizing the overlapping coverage. Reference [7] designs a novel reformulation for a probabilistic variant of the set covering problem as well as derives efficient bounds. Reference [8] proposes an efficient linearisation technique to handle the location problem with site-dependent failure probabilities. Reference [9] studies several omnidirectional probabilistic coverage models for automated external defibrillators (AEDs) in public areas. Reference [10] considers uncertainty in real world practice and presents an efficient under-approximation for the two-level cooperative robust set covering problem. We note that the coverage uncertainty is considered and approximated in the constraints in [10], while we regard it as the objective function. Reference [11], [12] models probabilistic angular coverage and designs meta-heuristic algorithms for solving it due to its computational complexity. However, these algorithms provide only empirically good solutions and are not theoretically exact.

Hence, in this paper, we jointly consider angular coverage and coverage under uncertainty and propose the angular coverage under uncertainty (ACU) model. First, a compact binary integer programming (IP) [13] formulation is designed. The formulation aims to minimize the weighted non-coverage probability over the target set with practical co-location and budget constraints. Co-location indicates that multiple sensors (e.g., surveillance cameras) with various directions can be deployed in the same location. Second, recognizing the non-linearity, non-convexity, and non-separability of the basic formulation, we introduce the reformulation-linearisation technique (RLT) to reformulate a tractable linear model. Third, the obtained linear model is solved via a customized branch-and-cut (B&C) algorithm. It provides a

This work was supported in part by the National Natural Science Foundations of China (NSFC) under Grant 62273044, 62022015, and 62088101, and in part by National Natural Science Foundation of China National Science Fund for Distinguished Young Scholars 62025301. Shuxin Ding is partially supported by the Young Elite Scientist Sponsorship Program by China Association for Science and Technology (CAST) under Grant 2022QNRC001 and the Youth Talent Program Supported by China Railway Society. *(Corresponding author: Chen Chen.)*

[1]National Key Lab of Autonomous Intelligent Unmanned Systems, Beijing Institute of Technology, Beijing 100081, China xiaofan@bit.edu.cn

[2]Beijing Institute of Technology Chongqing Innovation Center, Chongqing, 401120, China

[3]Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China, and the Traffic Management Laboratory for High-Speed Railway, National Engineering Research Center of System Technology for High-Speed Railway and Urban Rail Transit, Beijing 100081, China.

tight lower bound for the reformulated problem, which is also proven to be the optimal value for the original problem. The proposed ACU is analysed over instances ranging from moderate to large size scaling up to 4,000 dimensional decision variables.

The rest of the paper is organized as follows. Section II formulates the proposed ACU. Section III introduces the RLT and illustrates the exact algorithm procedure. Computational experiments and analysis are presented in section IV. Section V concludes this paper and discusses some open issues.

II. Formulating the ACU

In ACU, sensors $j \in \mathcal{N} = \{1, \ldots, n\}$ are deployed to provide coverage for the targets $i \in \mathcal{M} = \{1, \ldots, m\}$, where n denotes the number of candidate positions for sensors and m denotes the number of targets. Each target is associated with a weight w_i. Considering the direction of the angular sensor, we divide the circular FoV equally according to the parameter $k \in \mathcal{P} = \{1, \ldots, p\}$ as shown in Fig. 1. Accordingly, let p_{ijk} denote the target i is covered by j-th sensor with k-th sector of FoV. To make the mathematical model compact, we define $j, k \in \mathcal{T} = \mathcal{N} \times \mathcal{P}$ where $\times$ denotes the Cartesian product. For each target, the coverage by various sensors is independent under the assumption. Thus, the non-coverage probability of target $i \in \mathcal{M}$ by the coverage set is $\prod_{j \in \mathcal{N}, k \in \mathcal{P}} (1 - p_{ijk})$. Define for all $i \in \mathcal{M}$, $\mathcal{T}_i \equiv \{j, k \in \mathcal{T} : 0 < p_{ijk} \leq 1\}$, $\mathcal{T}_i^+ \equiv \{j, k \in \mathcal{T} : 0 < p_{ijk} < 1\}$, and $\mathcal{T}_i^1 \equiv \mathcal{T}_i - \mathcal{T}_i^+ = \{j, k \in \mathcal{T} : p_{ijk} = 1\}$.

The sensing model is modified from that in reference [9] and depicted for all $i \in \mathcal{M}, j \in \mathcal{N}, k \in \mathcal{P}$ as follows

$$p_{ijk} = \begin{cases} 1 \cdot \mathbf{I}(a_{ij}, k), & \text{if } d_{ij} \leq d_1, \\ e^{-\alpha(d_{ij} - d_1)} \cdot \mathbf{I}(a_{ij}, k), & \text{if } d_1 < d_{ij} \leq d_2, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where d_{ij} denotes the Euclidean distance metric between sensor j and target i. $\mathbf{I}(\cdot)$ is the indicator function. If the angle a_{ij} between sensor j and target i falls within the k-th sector of FoV, the indicator function value is 1. Otherwise, the indicator function value is 0. α denotes the decay factor. d_1 and d_2 can be determined by the performance parameter of specified sensors.

The basic formulation of ACU is presented as

$$\min_{\boldsymbol{x}} \sum_{i \in \mathcal{M}} w_i \prod_{j,k \in \mathcal{T}_i} (1 - p_{ijk})^{x_{jk}}, \tag{2}$$

$$s.t. \sum_{k \in \mathcal{P}} x_{jk} \leq C, \ \forall j \in \mathcal{N}, \tag{2a}$$

$$\sum_{j,k \in \mathcal{T}} x_{jk} = P, \tag{2b}$$

$$x_{jk} \in \{0, 1\}, \ \forall j \in \mathcal{N}, k \in \mathcal{P}. \tag{2c}$$

where the objective function seeks to minimize the weighted non-coverage probability over the target set. Eq. (2a) depicts the co-location. Co-location is not allowed if $C = 1$. Otherwise, C sensors with various directions can be deployed in the same candidate position. Eq. (2b) is a budget constraint that denotes that the number of the deployed sensors is restricted to P. Eq. (2c) indicates the binary decision variables. x_{jk} equals to 1 if the sensor is deployed in the j-th candidate position with k-th sector of FoV, and 0 otherwise.

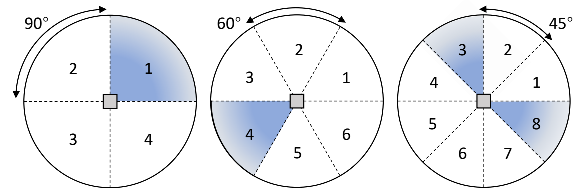


Fig. 1. Divisions of the circular FoV for 90° ($p = 4$), 60° ($p = 6$), and 45° ($p = 8$), respectively. Note that under the circumstances of $p = 8$, two sensors are co-located in the 3-rd and 8-th sector of the same position.

Moreover, with the pre-defined $\mathcal{T}_i$, $\mathcal{T}_i^+$ and $\mathcal{T}_i^1$, we can rewrite ACU in the form as follows

$$\min_{\boldsymbol{x}} \sum_{i \in \mathcal{M}} w_i \left[\prod_{j,k \in \mathcal{T}_i^1} (1 - x_{jk}) \prod_{j,k \in \mathcal{T}_i^+} (1 - p_{ijk})^{x_{jk}} \right], \tag{3}$$

$$s.t. \sum_{k \in \mathcal{P}} x_{jk} \leq C, \ \forall j \in \mathcal{N}, \tag{3a}$$

$$\sum_{j,k \in \mathcal{T}} x_{jk} = P, \tag{3b}$$

$$x_{jk} \in \{0, 1\}, \ \forall j \in \mathcal{N}, k \in \mathcal{P}. \tag{3c}$$

We can make a preliminary analysis of the above mathematical model. The formulation of ACU is a binary IP problem with linear constraints. However, the objective function is non-linear, non-convex, and non-separable, making it computationally intractable for traditional exact algorithms. We exploit the mathematical structure of the problem and apply RLT to derive a tractable model in the following section. We also customize a B&C algorithm for solving the derived problem in an exact manner.

III. RLT and Exact Algorithm

We overview the solution methods for ACU in Fig. 2.

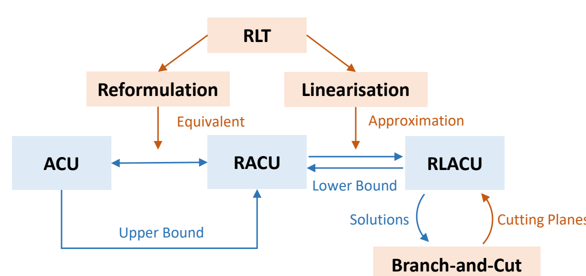


Fig. 2. Overview of the solution methods for ACU. Note that the blue part corresponds to the mathematical model, while the red part corresponds to the solution approaches.

Recap that the non-linearity, non-convexity, and non-separability make ACU computationally intractable. Thus, we focus on applying modelling techniques along with efficient algorithms to solve it. First, we reformulate ACU and acquire an equivalent model RACU which is proved in the following subsection. Considering the non-linearity of RACU, we further linearise it to obtain a mixed-integer linear programming model RLACU. It is interesting that RLACU is a relaxation of RACU and provides a lower bound for RACU (i.e., also for its equivalent model ACU). Moreover, RACU itself can recover the upper bound for ACU. *If we approximate RACU tightly with RLACU, we can let the gap between the upper bound and lower bound be zero, thus obtaining the optimal solution for ACU*. Subsequently, a B&C algorithm is customized to solve RLACU exactly. The B&C algorithm detects the current solution along with the mathematical structure, and generate a set of cutting planes to enforce a tighter approximation of RACU.

We illustrate the RLT, bounding scheme, and B&C algorithm sequentially in the following subsection.

A. RLT

RLT is a useful technique helping solve a wide variety of computationally hard IP formulations that arise in real world practice [14], [15]. In this paper, we introduce the RLT to yield a linear model. The optimal value of the linear model provides a tight lower bound for the optimal value to the original binary IP problem [7], [16]. We also observe that the linear model itself solves the underlying IP problem.

In addition, for each $i \in \mathcal{M}$, we define two auxiliary continuous variables ξ_i and η_i, and $\mu_{ijk} = -ln(1-p_{ijk})$ for $\forall j,k \in \mathcal{T}_i^+, i \in \mathcal{M}$. We present the RACU, the reformulation of the original ACU as follows

$$\min_{\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{x}} \sum_{i\in\mathcal{M}} w_i\xi_i, \tag{4}$$

$$s.t.\ \xi_i \le 1 - x_{jk},\ \forall j,k \in \mathcal{T}_i^1, i \in \mathcal{M}, \tag{4a}$$

$$\xi_i \ge \eta_i - \sum_{j,k\in\mathcal{T}_i^1} x_{jk},\ \forall i \in \mathcal{M}, \tag{4b}$$

$$\eta_i = e^{-\sum_{j,k\in\mathcal{T}_i^+}\mu_{ijk}x_{jk}},\ \forall i \in \mathcal{M}, \tag{4c}$$

$$\sum_{k\in\mathcal{P}} x_{jk} \le C,\ \forall j \in \mathcal{N}, \tag{4d}$$

$$\sum_{j,k\in\mathcal{T}} x_{jk} = P, \tag{4e}$$

$$\xi_i, \eta_i \ge 0,\ \forall i \in \mathcal{M}, \tag{4f}$$

$$x_{jk} \in \{0,1\},\ \forall j \in \mathcal{N}, k \in \mathcal{P}. \tag{4g}$$

Theorem 1: Model RACU is equivalent to model ACU if $(\bar{\xi},\bar{\eta},\bar{x})$ solves RACU, then $\bar{x}$ solves ACU and obtains the identical objective value. What is more, if there is an optimal solution for ACU, then RACU also has its optimum, with the identical objective value.

Proof: (i) We prove that if $(\bar{\xi},\bar{\eta},\bar{x})$ solves RACU, $\bar{x}$ can solve ACU. Let $(\bar{\xi},\bar{\eta},\bar{x})$ solves RACU, we can derive that $\bar{x}$ is feasible to ACU for $\bar{x}$ satisfies Eq. (4d), (4e), and (4g).

(ii) We prove that the solution $(\bar{\xi},\bar{\eta},\bar{x})$ of RACU and the solution $\bar{x}$ of ACU has the identical objective value. Define $\bar{\Omega}_i = \prod_{j,k\in\mathcal{T}_i^1}(1-\bar{x}_{jk})\prod_{j,k\in\mathcal{T}_i^+}(1-p_{ijk})^{\bar{x}_{jk}}$ for all $i \in \mathcal{M}$. We utilize the structure of $\mathcal{T}_i^1$ to complete this proof. Consider any $i \in \mathcal{M}$. If $\bar{x}_{jk} = 1$ for any $j,k \in \mathcal{T}_i^1$, then $\bar{\Omega}_i = 0$. Moreover, Eq. (4a) and (4f) derives $\bar{\xi}_i = 0$. Note that $\bar{\eta}_i \le 1$ from Eq. (4c) since $\mu_{ijk} = -ln(1-p_{ijk})>0$ for $\forall j,k \in \mathcal{T}_i^+$. Also, Eq. (4b) is satisfied under $\bar{\xi}_i$ and $\bar{\eta}_i$. Otherwise, if $\bar{x}_{jk} = 0$ for all $j,k \in \mathcal{T}_i^1$, then $\bar{\eta}_i \le \bar{\xi}_i \le 1$ according to Eq. (4a) and (4b). Since the objective function minimizes $\bar{\xi}_i$, $\bar{\eta}_i = \bar{\xi}_i = e^{-\sum_{j,k\in\mathcal{T}^+}\mu_{ijk}\bar{x}_{jk}} = \prod_{j,k\in\mathcal{T}_i^+}(1-p_{ijk})^{\bar{x}_{jk}} = \bar{\Omega}_i$. Overall, we complete this part.

(iii) We prove that if any x^* is feasible for ACU, (ξ^*,η^*,x^*) exists and is feasible to RACU. They have identical objective value. The proposed problem can be solved optimally if and only if it is feasible. Consider setting η^* given $x = x^*$ using Eq. (4c). For each $i \in \mathcal{M}$, if $x^*_{jk} = 0$ for all $j,k \in \mathcal{T}_i^1$, set $\xi_i^* = \eta_i^*$; if $x^*_{jk} = 1$ for any $j,k \in \mathcal{T}_i^1$, set $\xi_i^* = 0$. Based on the proof in (ii), we can conclude that the feasibility of (ξ^*,η^*,x^*) to RACU. Also, the identical objective value is achieved. This completes our proof. ■

We have reformulated the problem RACU. However, Eq. (4c) introduces the non-linear constraints which are unfriendly to mixed-integer programming. We linearise Eq. (4c) using a set of tangency functions

$$\eta_i \le 1 - \frac{(1-e^{-\mu_i})}{\mu_i}\beta_i, \forall i \in \mathcal{M}, \tag{5}$$

$$\begin{aligned} &\eta_i \ge e^{-\bar{\beta}_i} - (\beta_i - \bar{\beta}_i)e^{-\bar{\beta}_i}, \\ &\quad \forall\bar{\beta}_i \in \left\{0, \frac{1}{t}\mu_i, \cdots, \frac{t-1}{t}\mu_i, \mu_i\right\}, \forall i \in \mathcal{M}, \end{aligned} \tag{6}$$

where for all $i \in \mathcal{M}$, $\beta_i = \sum_{j,k\in\mathcal{T}_i^+}\mu_{ijk}x_{jk} \le \mu_i$ and $\mu_i = \sum_{j,k\in\mathcal{T}_i^+}\mu_{ijk}$. t is an pre-defined integer and $t \ge 2$. We can observe from Fig. 3 that Eq. (5) and (6) construct a convex polyhedral envelope which approximates Eq. (4c). The convex polyhedral envelope provides a tighter approximation as t increases. By substituting Eq. (5) and (6) into RACU, we derive a linear formulation RLACU

$$\min_{\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{x}} \sum_{i\in\mathcal{M}} w_i\xi_i, \tag{7}$$

$$s.t.\ \xi_i \le 1 - x_{jk},\ \forall j,k \in \mathcal{T}_i^1, i \in \mathcal{M}, \tag{7a}$$

$$\xi_i \ge \eta_i - \sum_{j,k\in\mathcal{T}_i^1} x_{jk},\ \forall i \in \mathcal{M}, \tag{7b}$$

$$\eta_i \le 1 - \frac{(1-e^{-\mu_i})}{\mu_i}\sum_{j,k\in\mathcal{T}_i^+}\mu_{ijk}x_{jk},\ \forall i \in \mathcal{M}, \tag{7c}$$

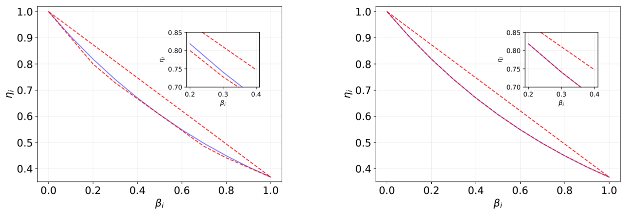
$$\begin{aligned} &\eta_i \ge e^{-\bar{\beta}_i} - \left(\sum_{j,k\in\mathcal{T}_i^+}\mu_{ijk}x_{jk} - \bar{\beta}_i\right)e^{-\bar{\beta}_i}, \\ &\quad \forall\bar{\beta}_i \in \left\{0, \frac{1}{t}\mu_i, \cdots, \frac{t-1}{t}\mu_i, \mu_i\right\}, \forall i \in \mathcal{M}, \end{aligned} \tag{7d}$$

$$\sum_{k\in\mathcal{P}} x_{jk} \le C,\ \forall j \in \mathcal{N}, \tag{7e}$$

$$\sum_{j,k\in\mathcal{T}} x_{jk} = P, \tag{7f}$$

$$\xi_i, \eta_i \ge 0,\ \forall i \in \mathcal{M}, \tag{7g}$$

$$x_{jk} \in \{0,1\},\ \forall j \in \mathcal{N}, k \in \mathcal{P}. \tag{7h}$$



(a) $t = 2$ (b) $t = 10$

Fig. 3. The blue curve depicts the compact form of Eq. (4c) which is $\eta_i = e^{-\beta_i}$. The red curves representing Eq. (5) and (6) which construct a convex polyhedral envelope that approximates the blue curve. As the value of the integer t increases, the convex polyhedral envelope provides a tighter approximation.

We briefly summarize that there mathematical models are presented, including basic ACU and the derived RACU and RLACU. We have proved that RACU is equivalent to ACU. The linear RLACU is an approximation and also a relaxation of RACU. We tend to approximate RACU tightly with RLACU so that we can let the gap between the upper bound and lower bound of ACU be zero. Thus, the optimality is obtained. In the following subsection, we present the bounding scheme and exact algorithm for solving RLACU.

B. Bounding Scheme

Recap that ACU is equivalent to RACU. On the one hand, if x^* can solve RLACU optimally, then it is also feasible to ACU. We recover the upper bound of ACU via its objective value $\sum_{i \in \mathcal{M}} w_i \prod_{j,k \in \mathcal{T}_i} (1 - p_{ijk})^{x^*_{jk}} = \sum_{i \in \mathcal{M}} w_i \prod_{j,k \in \mathcal{T}_i} (1 - p_{ijk} x^*_{jk})$. This upper bound is available for RACU. On the other hand, RLACU provides a lower bound for RACU (i.e., also for ACU). If we approximate RACU tightly with RLACU, we can let the gap between the upper bound and lower bound be zero for optimality proof.

C. B&C Algorithm

B&C [17] is the mainstream framework for solving IP problems. B&C is the intensification of the branch-and-bound algorithm. That is, B&C involves applying branch-and-bound algorithm [18] and using cutting planes [19], [20], [21] to tighten the linear programming relaxation. B&C is the cornerstone of the IP solution approach embedded in modern commercial solvers [22].

RLACU is a linear binary mixed-integer programming problem and itself is solved via B&C first. Note that RLACU is obtained using RLT which provides a tight lower bound for the optimal value to the binary IP problem. To enforce the lower bound tighter, we sequentially detect the current solution along with the mathematical structure, and generate a set of customized cutting planes (i.e., constraints) efficiently.

$$\eta_i \geq e^{-\beta_i^*} - \left(\sum_{j,k \in \mathcal{T}_i^+} \mu_{ijk} x_{jk} - \beta_i^* \right) e^{-\beta_i^*} \tag{8}$$

Algorithm 1 shows the pseudo-code of the B&C algorithm procedure. RLACU is first solved optimally with loaded parameters. Then, the constraints violation is detected, and

Algorithm 1: B&C Algorithm

Input: Hyper-parameters $\{t, \epsilon\}$ where ϵ denotes the lower bound improvement tolerance, RLACU;
Output: Solution to RLACU.
// Load Parameters and Solve RLACU
Solve RLACU and denote the obtained optimal solution as (ξ^*, η^*, x^*);
Initialize ρ_i with a negative value (e.g., -1);
if $\rho_i < 0$ *for any* $i \in \mathcal{M}$*, or the lower bound is more than* ϵ *better than that in the previous loop* **then**
  // Identify the Cutting Planes
  Compute $\beta_i^* = \sum_{j,k \in \mathcal{T}_i^+} \mu_{ijk} x^*_{jk}$ for each $i \in \mathcal{M}$ and $\rho_i = \eta_i^* - e^{-\beta_i^*}$. We customize the cut of form Eq. (8) with β_i^*;
  // Add the Cutting Planes
  Add the identified cut sequentially $\forall i \in \mathcal{M}$ for the current RLACU model;
  Solve RLACU for the new optimal solution;
end
Return the solution to RLACU;

the cutting planes are generated using Eq. (8) with the current solution. RLACU is solved again after adding the above cutting planes sequentially. Repeat these procedures until the stopping criterion is met. That is, all the violated cuts are identified and the iterated lower bound is not more than ϵ better than the bound in the previous loop.

IV. Computational Experiments and Analysis

In this section, we present the design of experiment and the main results of the proposed approach as well as conduct empirical analysis on the hyper-parameters to provide some operational insights for practitioners.

A. Experiment Settings

We generate a set of problem instances ranging from moderate to large scale as follows. The spatial coordinates of both sensors and targets are generated randomly within the 1×1 grid. The weight of each target is computed by sampling from the uniform distribution in $[0, 1]$. We test parameter P as $\lceil 0.1 \times n \rceil$, $\lceil 0.2 \times n \rceil$, $\lceil 0.5 \times n \rceil$ and C as $1, 3$ in the ACU model. Recap that C depicts the co-location. P is the restriction of the number of the deployed sensors and n denotes the number of the candidate positions for sensors. The setting of P indicates the relatively low, median, and high resource availability. The decay factor is $\alpha = 0.05$ as that in [9]. The performance parameters of sensors d_1 and d_2 corresponding to each P parameter are 0.2 and 0.4, 0.1 and 0.2, 0.04 and 0.08 if $n \leq 50$, respectively. If $n > 50$, each performance parameter is multipled by $\frac{100}{n}$. Each instance in each case is evaluated five times randomly. t and ϵ are set as 10 and 0, respectively, while we conduct some analysis on these hyper-parameters in the following subsection.

All experiments are done on a computer with an Intel(R) Core(TM) i9-9900 CPU @ 3.10GHz with 64GB of RAM.

TABLE I
COMPUTATIONAL RESULTS FOR ACU

Case	n	m	p	$P = \lceil 0.1 \times n \rceil$				$P = \lceil 0.2 \times n \rceil$				$P = \lceil 0.5 \times n \rceil$			
				$C = 1$		$C = 3$		$C = 1$		$C = 3$		$C = 1$		$C = 3$	
				Time	% Opt	Time	% Opt	Time	% Opt	Time	% Opt	Time	% Opt	Time	% Opt
1	15	15	6	0.67	100.00	0.68	100.00	0.76	100.00	0.73	100.00	0.94	100.00	0.66	100.00
2	15	15	8	0.93	100.00	0.88	100.00	0.99	100.00	1.03	100.00	0.86	100.00	0.84	100.00
3	20	20	6	1.15	100.00	1.26	100.00	1.31	100.00	1.26	100.00	1.14	100.00	1.15	100.00
4	20	20	8	1.59	100.00	1.59	100.00	1.61	100.00	1.80	100.00	1.62	100.00	1.52	100.00
5	30	30	6	2.74	100.00	2.83	100.00	2.57	100.00	2.78	100.00	2.63	100.00	2.35	100.00
6	30	30	8	3.76	100.00	3.39	100.00	3.86	100.00	3.57	100.00	3.44	100.00	3.43	100.00
7	50	50	6	7.66	100.00	7.71	100.00	7.33	100.00	8.17	100.00	8.43	100.00	7.84	100.00
8	50	50	8	9.24	100.00	8.71	100.00	9.50	100.00	9.37	100.00	9.81	100.00	11.47	100.00
9	50	100	6	14.94	100.00	14.44	100.00	15.67	100.00	15.44	100.00	15.36	100.00	13.97	100.00
10	50	100	8	18.81	100.00	20.35	100.00	19.47	100.00	19.56	100.00	20.23	100.00	20.17	100.00
11	100	100	6	38.76	100.00	34.62	100.00	30.37	100.00	30.04	100.00	30.63	100.00	29.05	100.00
12	100	100	8	38.96	100.00	39.27	100.00	36.07	100.00	38.46	100.00	38.23	100.00	37.62	100.00
13	200	200	6	135.25	100.00	121.68	100.00	114.88	100.00	113.75	100.00	116.57	100.00	155.76	100.00
14	200	200	8	154.56	100.00	145.90	100.00	155.70	100.00	149.52	100.00	143.84	100.00	199.39	100.00
15	200	500	6	281.78	100.00	291.81	100.00	269.64	100.00	297.63	100.00	282.48	100.00	289.37	100.00
16	200	500	8	358.61	100.00	343.38	100.00	343.97	100.00	352.53	100.00	381.89	100.00	376.48	100.00
17	500	500	6	666.87	100.00	756.40	100.00	716.87	100.00	785.84	100.00	744.14	100.00	650.73	100.00
18	500	500	8	887.78	100.00	934.12	100.00	935.90	100.00	1079.55	100.00	877.21	100.00	794.43	100.00
19	500	1000	6	1553.81	100.00	1779.28	100.00	1388.49	100.00	1404.63	100.00	1745.27	100.00	1651.59	100.00
20	500	1000	8	1782.49	100.00	1797.68	100.00	1771.82	100.00	1778.03	100.00	1771.75	100.00	1816.93	100.00

Remark: Each instance in each case is evaluated five times randomly, and the statistical results are collected and presented.

The B&C algorithm is implemented using Gurobi Optimizer [22] with default parameter settings in Python 3.7 environment.

B. Result, Analysis, and Operational Insights

The main results are shown in Table I with the following statistical metrics

- Time: The arithmetic mean CPU run time in seconds.
- % Opt: Defined as (lower bound value)/(upper bound value)$\times 100$, where the lower bound is the objective value of the solution obtained by RLACU. The upper bound value is the objective value of the solution for the original ACU. Note that $\%$ Opt is also the arithmetic mean value for the repeated five times experiments. If $\%$ Opt equals 100, the optimality of ACU is achieved.

We can conclude from Table I that ACU can be solved optimally at scale using the proposed approach. We observe that the optimal results are not sensitive to parameters P and C. That is, ACU can be utilized to handle various co-location requirements and different resource availability. $n \times p$ presents the dimension of decision variables. As the dimension increases, we witness the growing computational time. We report the % Opt in Table I and discuss additional issues on the larger-scale instances. We notice that most of the optimization time is occupied during the first step and the number of cuts introduced is relatively stable verified in Table II. To solve larger-scale optimization problems over thousands of dimensions, on the one hand, we can allow more CPU time (i.e., over 1800.00 CPU time in seconds) on parallel multi-core machines. On the other hand, the problem in the first step can be accelerated by heuristic initialization with high-quality feasible solutions for the commercial solver.

We also conduct empirical analysis on the hyper-parameters of the algorithm including t and ϵ. For t, we select moderate-sized case 11 to conduct an experiment with $P = \lceil 0.1 \times n \rceil$ and $C = 1$. Recap that t is an pre-defined integer and $t \geq 2$, we let t range in $[2, 5, 10, 50, 100, 200, 500]$. It is shown from Table II that as t increases, the CPU time rises due to the complexity of the model solved in the first step. However, for the moderate-sized case 11, these t always solve the instances optimally. What is more, the change in the number of customized cuts is not significant. Thus, a relatively small t such as 10 is enough to produce tight bound effectively as well as balance the computational complexity of the mathematical model.

TABLE II
PARAMETER ANALYSIS ON t FOR ACU. MODERATE-SIZED CASE 11 IS SELECTED WITH $P = \lceil 0.1 \times n \rceil$ AND $C = 1$.

Metric	$t = 2$	$t = 5$	$t = 10$	$t = 50$	$t = 100$	$t = 200$	$t = 500$
Time	20.46	22.71	31.60	101.47	183.67	355.93	863.54
% Opt	100.0	100.00	100.00	100.00	100.00	100.00	100.00
Cuts	700	600	600	700	600	600	600

For ϵ, in real world applications for coverage under uncertainty, a considerable portion of instances are budget limited. We note for practitioners that under the above circumstance, a relatively large ϵ (e.g., 0.001) may result in faster convergence. However, sometimes complete coverage can be achieved (i.e., the objective value of ACU is zero) and the values of the bounds are tiny. We suggest that the value ϵ is set as 0 to solve these instances optimally and stably.

V. CONCLUSION AND DISCUSSION

In this paper, we have presented an IP approach for ACU problem. ACU jointly models angular coverage and coverage under uncertainty, as well as takes practical constraints such as co-location and limited budget into consideration. An exact B&C algorithm coupled with RLT is designed to solve this problem at scale. The mathematical structure is detected and problem-specific cutting planes are customized. With efficient parallel multi-core machines and acceleration techniques for solvers, our proposed approach may solve more than thousands of dimensional decision variables. We also make some analyses on the parameter settings to provide operational insights for practitioners.

In the future, we propose to integrate more practical issues in the ACU, including heterogeneous sensors and 3-D environment. Take an example, in a 3-D environment, both the pan angle and tilt angle of sensors should be considered. Our proposed ACU is compact enough for these practical issues since the discretized tilt angles or heterogeneous set can be wrapped in the set $\mathcal{T}$. Also, we suggest the design of suitable heuristic algorithms [23] to near-optimally solve the larger-scale ACU problem. Meanwhile, the proposed model and approach is useful for tackling a series of real world tasks [24], [25] in the field of location analysis, spatial optimization, urban operations, wireless communications, healthcare, etc.

REFERENCES

[1] R. L. Church, A. Murray *et al.*, "Location Covering Models," *Advances in Spatial Science*, 2018.

[2] S. Ding, C. Chen, Q. Zhang, B. Xin, and P. M. Pardalos, *Metaheuristics for Resource Deployment under Uncertainty in Complex Systems*. CRC Press, 2021.

[3] R. Church and C. ReVelle, "The maximal covering location problem," in *Papers of the Regional Science Association*, vol. 32, no. 1. Springer-Verlag Berlin/Heidelberg, 1974, pp. 101–118.

[4] C. Toregas, R. Swain, C. ReVelle, and L. Bergman, "The location of emergency service facilities," *Operations Research*, vol. 19, no. 6, pp. 1363–1373, 1971.

[5] E. Yildiz, K. Akkaya, E. Sisikoglu, and M. Y. Sir, "Optimal camera placement for providing angular coverage in wireless video sensor networks," *IEEE Transactions on Computers*, vol. 63, no. 7, pp. 1812–1825, 2013.

[6] A. M. Heyns, "Optimisation of surveillance camera site locations and viewing angles using a novel multi-attribute, multi-objective genetic algorithm: A day/night anti-poaching application," *Computers, Environment and Urban Systems*, vol. 88, p. 101638, 2021.

[7] Y. Lee, H. D. Sherali, I. Kwon, and S. Kim, "A new reformulation approach for the generalized partial covering problem," *Naval Research Logistics*, vol. 53, no. 2, pp. 170–179, 2006.

[8] J. R. O'Hanley, M. P. Scaparra, and S. García, "Probability chains: A general linearization technique for modeling reliability in facility location and related problems," *European Journal of Operational Research*, vol. 230, no. 1, pp. 63–75, 2013.

[9] T. C. Chan, D. Demirtas, and R. H. Kwon, "Optimizing the deployment of public access defibrillators," *Management Science*, vol. 62, no. 12, pp. 3617–3635, 2016.

[10] S. Ding, Q. Zhang, and Z. Yuan, "An under-approximation for the robust uncertain two-level cooperative set covering problem," in *IEEE Conference on Decision and Control*. IEEE, 2020, pp. 1152–1157.

[11] A. S. R. S. Benyattou, "Toward a realistic approach for the deployment of 3D wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 4, pp. 1508–1519, 2022.

[12] T. Wu, Y. Zhang, C. Miao, C. Chen, and S. Ding, "Mixed-variable correlation-aware metaheuristic for deployment optimization of 3-D sensor networks," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2024, pp. 1390–1398.

[13] D.-S. Chen, R. G. Batson, and Y. Dang, *Applied Integer Programming: Modeling and Solution*. John Wiley & Sons, 2011.

[14] W. P. Adams and H. D. Sherali, "A hierarchy of relaxations leading to the convex hull representation for general discrete optimization problems," *Annals of Operations Research*, vol. 140, pp. 21–47, 2005.

[15] H. D. Sherali and W. P. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Springer Science & Business Media, 2013, vol. 31.

[16] H. D. Sherali and A. Alameddine, "A new reformulation-linearization technique for bilinear programming problems," *Journal of Global Optimization*, vol. 2, pp. 379–410, 1992.

[17] J. E. Mitchell, "Branch-and-cut algorithms for combinatorial optimization problems," *Handbook of Applied Optimization*, 2002.

[18] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations Research*, vol. 14, no. 4, pp. 699–719, 1966.

[19] E. Balas, S. Ceria, G. Cornuéjols, and N. Natraj, "Gomory cuts revisited," *Operations Research Letters*, vol. 19, no. 1, pp. 1–9, 1996.

[20] F. W. Kong, D. Kuhn, and B. Rustem, "A cutting-plane method for mixed-logical semidefinite programs with an application to multi-vehicle robust path planning," in *IEEE Conference on Decision and Control*. IEEE, 2010, pp. 1360–1365.

[21] A. Testa, A. Rucco, and G. Notarstefano, "A finite-time cutting plane algorithm for distributed mixed integer linear programming," in *IEEE Annual Conference on Decision and Control*. IEEE, 2017, pp. 3847–3852.

[22] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2022. [Online]. Available: https://www.gurobi.com

[23] E.-G. Talbi, *Metaheuristics: From Design to Implementation*. John Wiley & Sons, 2009.

[24] S. He, K. Shi, C. Liu, B. Guo, J. Chen, and Z. Shi, "Collaborative sensing in internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1435–1474, 2022.

[25] J. Chen, J. Sun, and G. Wang, "From unmanned systems to autonomous intelligent systems," *Engineering*, vol. 12, pp. 16–19, 2022.