IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 March 2024, accepted 10 June 2024, date of publication 19 June 2024, date of current version 28 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3416871

RESEARCH ARTICLE

The Angular Set Covering Problem

FREDY BARRIGA-GALLEGOS[1], **ARMIN LÜER-VILLAGRA**[2], **AND GABRIEL GUTIÉRREZ-JARPA**[1]

[1]School of Industrial Engineering, Pontificia Universidad Católica de Valparaíso, Valparaíso 2340025, Chile
[2]Department of Engineering Sciences, Universidad Andres Bello, Talcahuano 4260000, Chile

Corresponding author: Fredy Barriga-Gallegos (fredy.barriga@pucv.cl)

This work was supported in part by Fondo Nacional de Desarrollo Científico y Tecnológico (FONDECYT) under Grant 1231023 and in part by the Agencia Nacional de Investigación y Desarrollo (ANID)-Programa Formación de Capital Humano Avanzado (PFCHA)/Doctorado Nacional under Grant 2023-21230628.

ABSTRACT We present an innovative extension of the Set Covering Problem, transitioning from a traditional radial covering to an angular covering structure. The decisions are based on locating the facilities and identifying the directional servers installed in each, covering a set of points in a geographic area. The strategic placement of surveillance cameras inspired this novel approach. We aim to minimize the cost of covering demand points by strategically installing directional servers on facilities. We propose an integer linear model and an initial approach based on column generation decomposition. We conducted extensive computational experiments on different test instances, including a practical case study involving positioning security cameras for surveillance in Valparaíso, Chile. The results highlight the effectiveness of column generation in achieving optimal solutions or significantly improving solution quality compared with solving the model directly with standard optimization solvers, especially for larger instances. In particular, the column generation algorithm achieves up to a 67% improvement in solution quality compared to the optimization model for real-world size instances, demonstrating its practical applicability and potential for enhancing surveillance infrastructure design.

INDEX TERMS Angular covering, column generation, combinatorial optimization, location problem, set covering.

I. INTRODUCTION

The set covering problem (SCP) is one of the most studied location problems. The SCP finds the minimum-cost set of facilities from a finite discrete candidate set, covering all demand points of a geographic area, including the service level through the distance between facilities and demand points.

The SCP was first proposed by [1] to determine the minimum number of police officers required to keep a road network covered. Reference [2] proposed the first mathematical model for locating emergency facility services. Moreover, SCP has been applied to other fields, such as bus stops [3], tool selection in flexible manufacturing systems [4], vehicle path planning [5], location of recycling facilities [6], location of healthcare facilities [7], relocation of hierarchical emergency response facilities [8], nature reserve design [9], optimization of gas detectors in chemical plants [10], and wireless network design for uncrewed aerial vehicles (UAV) in disaster areas [11].

Some assumptions were made in the mathematical model for the SCP:

1) All demand points are equally important in receiving service, independent of their demand level.
2) All demand points must be covered by at least one located facility.
3) The coverage is radial. Therefore, a facility covers a point of demand if the distance between the facilities is less than or equal to the coverage radius.
4) All covered demand points received the same service level, which did not depend on their distance from the facility that covered them.
5) These facilities are uncapacitated and do not have a maximum number of demand points to cover.

Several extensions of SCP have been proposed. Dynamic SCP assumes that the parameters of the model can take

The associate editor coordinating the review of this manuscript and approving it for publication was Geng-Ming Jiang.

different values depending on the period within the planning horizon [12]. The Capacitated SCP assumes that facilities have a limited capacity to attend to demand points [13]. The Multiple Optimal SCP determines the optimal number of facilities to cover all demand points as a secondary objective that minimizes the distance between facilities and demand points [14]. The Fuzzy SCP determines the optimal covering degree by considering fuzzy coverage [15]. In [16], an edge-covering problem (ECP) is addressed in a fuzzy environment, and three models were proposed. The set k-covering problem guarantees coverage of each demand point at least k times [17]. In [18], maximizing the demand points covered at least twice was proposed as a secondary objective. In [19], three models were proposed: a) gradual cover models, where the covered demand depends on the proximity between the facility and the demand point. b) Cooperative cover model, which considers multiple facilities to address the demands of each point. c) Variable radius model with coverage radius depending on location cost. In [20], a continuous version of a covering model was proposed, including risk. The fuzzy concept introduces a degree of customer satisfaction based on the covering radius. The model determines the locations of facilities in a continuous space. In [21], the coverage radius is a decision variable and not an exogenous parameter of the model applied to the hub-covering problem. In [22], deterministic equivalence was proposed to include the uncertainty in the SCP based on an uncertainty distribution.

However, some applications consider servers with angular covering, such as security cameras, where the angle defines the image quality. This structure guarantees the coverage of zones where the demand points are located and avoids covering unnecessary spaces where there are no demand points or, when necessary, requires service to some geographic areas (security cameras).

This angular covering structure is beneficial when the cost is associated with establishing coverage. Therefore, it is crucial to consider this structure to cover areas with clustered demand points, leaving areas with no demand points or minor importance uncovered. This extends the classical SCP problem because, together with the location decision, we consider an angular covering decision.

Figure 1 shows an example of an angular covering structure. The nodes, shown in light gray circles, represent the demand points. Squares represent the candidate locations of a facility. The nodes where the facilities are located are shown in dark grey. Servers are installed on the located facilities and establish angular covering, represented by the gray area covering the demand points. Instead of radial coverage, three servers establish angular coverage from each facility. One group is configured at 90°, 60°, and 45°, and the other is configured at 30°, 45°, and 90°. By adopting this innovative approach, we efficiently cover areas with demand points, a significant advantage of the angular coverage structure. A radial coverage structure also covers the demand points; however, the area would be larger, covering zones without demand points, which would mean higher costs for establishing coverage toward the same demand points.

Studies have addressed the location of security cameras that adopt an angular covering structure. In [23], the authors addressed the first two-dimensional (2D) angular covering problem. The coverage area was divided into grids to be covered. They proposed a covering model that minimizes server installation costs by covering all grids. The model includes a covering matrix that indicates whether the server can cover the grid or not. Therefore, the model is identical to the SCP model. They considered one server type and multiple configurations, installing at most one server in each grid centroid, including an angle-distance relationship. They presented the optimization model. In [24], the authors studied a three-dimensional (3D) angular covering problem. The area was divided into grids, and the goal was to install the minimum number of servers to obtain the most coverage possible. They considered one server type and multiple server configurations. The optimization model is not presented, but they solve instances using a particle swarm optimization algorithm. In [25], a 2D angular covering problem was addressed by dividing the area into grids. They proposed an extension of the maximal covering location problem (MCLP), which includes a service-level component that guarantees a given level of coverage. They also considered multiple server types and configurations. The servers are installed at the grid centroid, and the covering is cooperative; multiple servers can cover a grid. The server type and configuration decisions are based on an algorithm that includes server interferences. In [26], a 2D angular covering problem was considered, with the study area divided into grids. The objective is to maximize grid coverage, regardless of the cost of installing the servers, where the grids can be partially covered. They considered multiple server types, configurations, and up to eight possible positions. They did not present the optimization model but solved instances with two different algorithms. In [27], an angular covering problem in two dimensions was presented where the coverage area was divided into nodes. He proposed an angular covering model for a problem called the angled maximal covering location problem. It extends the MCLP with angular covering characteristics. In addition, the objective function maximizes coverage overlap. He considered multiple server types and configurations with several coverage levels depending on the day or night. He does not present a model for the problem but solves instances using a genetic algorithm.

Several studies have proposed methods to improve the computational time for solving SCP and their extensions. In [28], the authors proposed diverse preprocessing techniques for SCP with conflict constraints, where the simultaneous selection of pairs of columns is prohibited. These techniques include row and column reduction and the introduction of valid inequalities. In [29], the presolving methods for partial SCP are presented. The objective is to minimize facility location costs while ensuring coverage of

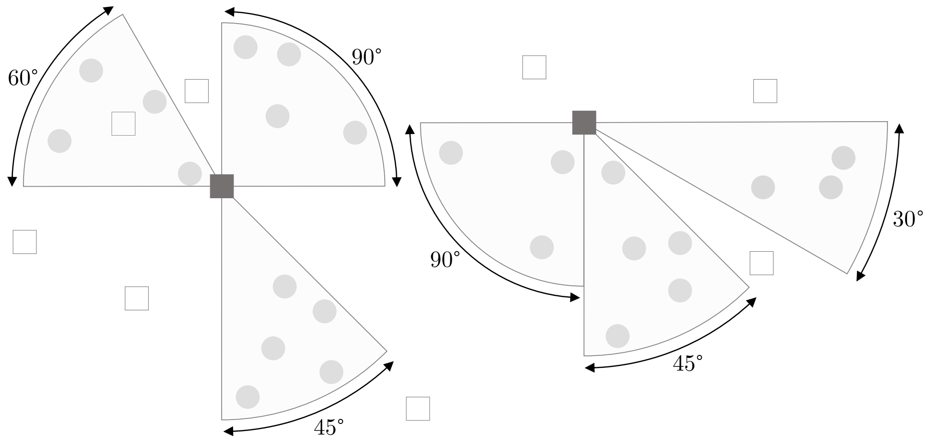

FIGURE 1. Example of the angular covering structure.

a specific number of demand points. In [30], a novel demand point aggregation method was proposed that effectively reduces the number of original demand points in covering problems. This method is accurate and significantly reduces computational costs in real-world location problems.

Exact methods, such as column generation, have been used in addition to the preprocessing and heuristic approaches mentioned above to solve location problems. The literature emphasizes the development of efficient methodologies for addressing covering problems using column generation. In [31], a column generation method was proposed that effectively solved the Maximal Covering Location Problem. In [32], a decomposition approach was introduced to solve the probabilistic maximal covering location-allocation problem, leveraging column generation and covering graph techniques. In [33], the authors presented a hybrid column generation approach to address large-scale coverage programs, particularly in transportation planning. In [34], the authors proposed extensions of column generation applied to greedy heuristics to address large-scale set covering problems.

Column generation is an effective method to solve complex coverage problems. This approach involves generating columns in each iteration to improve the solution quality. The relaxed master problem selects a subset of the available set of columns that converges to an optimal solution faster than the original model. It achieves this without enumerating all the columns but by generating only those considered favorable [35].

This research contributes to relaxing the third SCP assumption, leaving aside a radial covering structure to propose an alternative angular covering design where the facilities cover the specific zones that are the demand points or groups of demand points. We propose a mathematical formulation and model for the angular covering problem in two dimensions as an extension of the SCP, where the coverage area is represented by discrete demand points, considering multiple server types and angles. The model determines the server type, location, configuration, and installation position. In addition, it guarantees the coverage of all demand points, minimizes the total cost (facility and server), and allows the installation of multiple servers per location. In addition, a column generation algorithm is proposed to determine near-optimal solutions for the addressed problem. We solve a set of medium/large instances showing the performance of the model and the column generation methodology. Finally, we solve a real case to locate security cameras in Valparaíso, Chile.

The remainder of this paper is organized as follows. In section II, we formulate the angular set covering problem and present mathematical models considering an angular covering structure. Section III presents the first approach to column generation and the methodology based on it to determine a feasible solution. Section IV describes the computational experiments over the instance tests. In Section V, we apply our approach to a case study on the location of security cameras in Valparaíso, Chile. Finally, in Section VI, we present our conclusions and suggestions for future research. An outline of the acronyms used in this paper is showed in Table 1.

TABLE 1. Summary of acronyms.

SCP	set covering problem
MCLP	maximal covering location problem
CVRP	capacitated vehicle routing problem
ACP	angular set covering problem
CG	column generation

II. THE ANGULAR SET COVERING PROBLEM

This section presents the angular set covering problem (ACP). The angular covering structure simultaneously considers facility location and server installation decisions. Servers are installed in the facilities to establish the angular covering.

The ACP minimizes facility location and server installation costs, ensuring that at least one server covers each demand point. This model is a powerful tool for locating facilities and determining the server type, location, configuration (angle), and position (orientation).

The assumptions of ACP indicate that the facilities are selected from a set of candidate locations to be installed. The servers are installed on the facilities; that is, to install servers in a location, it is necessary to locate a facility. The covering distance and angle of the servers are deterministic. Thus, a demand point is covered if it is within the coverage area of the server. Both facilities and servers are uncapacitated. Therefore, facilities do not have a limit for installing servers, and servers do not have a limit for covering demand points. A demand point is served if it is covered by at least one server. Without loss of generality, the server positions are enumerated counterclockwise from the positive x-axis.

Fig. 2 shows the number of positions and their number. For server configurations of 90°, 60°, 45°, and 30°, the number of positions to install a server is 4, 6, 8, and 12, respectively. The number of positions changes depending on the angle, but they are numbered the same way up to cover 360°. When the angle associated with the server configuration is large, the number of positions is small. For example, there are four possible positions in the 90° server configuration, whereas in the 30° configuration, there are twelve possible positions.

Fig. 3 presents the two server types for the two configurations. For simplicity, the difference between the servers is the amount of covered area, which affects the covering distance for the same configuration. Fig. 3(a) shows the servers for a configuration of 90° where the covering distances are 11 and 16. Fig. 3(b) shows the same servers for a configuration of 45°. The covering distance of each server with a 45° configuration is greater than 90°, but the area covered is the same for each server.

A. MATHEMATICAL FORMULATION

Let M be the set of demand points, indexed by i. N is the set of candidate locations to locate a facility, indexed by j. T is the server configuration set that defines the feasible covering angles, indexed by t. For example $T = \{30°, 45°, 60°, 90°\}$. P_t is the set of positions for a server configuration t, e.g., $P_{60°} = \{1, 2, 3, 4, 5, 6\}$. Finally, S_t is the set of server types for a configuration t with different coverage area, e.g., $S_{60°} = \{1, 2\}$, meaning that there are two types of servers that cover 60°, as Fig. 2 shows. A server is identified as a quadruplet (t, s, p, j), which consists of configuration t where server type s is installed at position p at location j. We define facility location cost as g_j, which represents the cost of locating a facility at candidate location j. The server installation cost is c_s^t and represents the cost of installing server type s adopting server configuration t for a determined facility. Finally, the binary parameter α_{ijt}^{ps} is one if demand point i is covered by server (t, s, p, j), and zero otherwise.

The decision variables are:

$w_j = 1$, if and only if a facility is placed at location j, and 0 otherwise.

$y_{jt}^{ps} = 1$, if and only if a server (t, s, p, j) is installed, and 0 otherwise.

The integer linear model for the ACP is given as follows:

$$\min \quad \sum_{j \in N} g_j w_j + \sum_{j \in N} \sum_{t \in T} \sum_{p \in P_t} \sum_{s \in S_t} c_s^t y_{jt}^{ps} \tag{1}$$

$$\text{s.t.} \quad y_{jt}^{ps} \leq w_j, \quad \forall j \in N, t \in T, p \in P_t, s \in S_t, \tag{2}$$

$$\sum_{s \in S_t} y_{jt}^{ps} \leq 1, \quad \forall j \in N, t \in T, p \in P_t, \tag{3}$$

$$\sum_{j \in N} \sum_{t \in T} \sum_{p \in P_t} \sum_{s \in S_t} \alpha_{ijt}^{ps} y_{jt}^{ps} \geq 1, \quad \forall i \in M, \tag{4}$$

$$w_j \in \{0, 1\}, \quad \forall j \in N, \tag{5}$$

$$y_{jt}^{ps} \in \{0, 1\}, \quad \forall j \in N, t \in T, p \in P_t, s \in S_t. \tag{6}$$

Objective function (1) minimizes the total costs. The first term is the facility location costs, and the second is the server cost. Constraints (2) allow the installation of servers at locations where a facility is located. Constraints (3) ensure that at most, one type of server is installed in a position for a configuration considering each potential facility. Constraints (4) guarantee that all the demand points are covered by at least one server type. Constraints (5) and (6) define the domain of the decision variables.

As the size of instances to solve through the ACP model increases, the model can become intractable, requiring extensive computation time to find the optimal solution or obtain a low-quality feasible solution within a specified time interval. We propose using a Column Generation algorithm to address this challenge. This algorithm decomposes the ACP model into a master problem and subproblems. This approach aims to improve the performance of the algorithm, particularly in solving large-sized instances efficiently. The details of the column generation approach will be addressed in the next section.

III. A COLUMN GENERATION APPROACH

This section introduces a column generation approach for obtaining nearly optimal solutions for the ACP model. In the proposed Column Generation (CG) algorithm, we define a column as a combination of servers installed on a located facility. The master problem chooses combinations, and the subproblems are iteratively solved to propose new attractive combinations of servers (t, p, s) to improve the value of the master problem, allowing us to decompose the solution process.

A. THE MASTER PROBLEM

First, we formulate the problem regarding facility location and server combination decisions to obtain a mathematical formulation suitable for column generation. This formulation is called the following Master Problem.

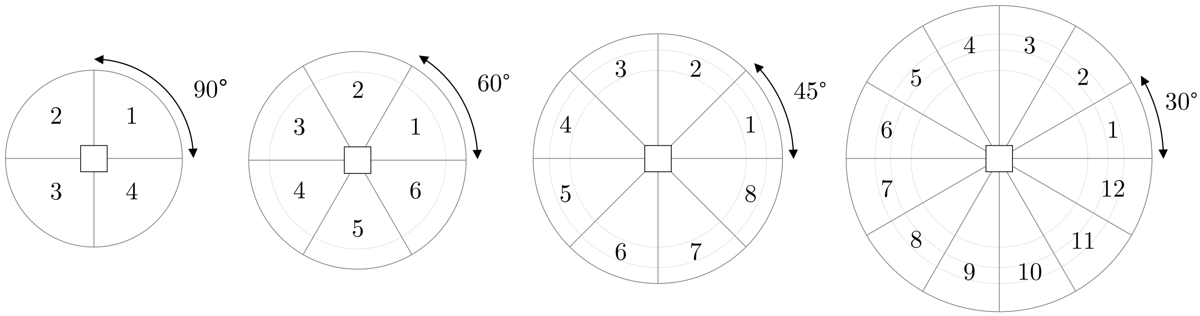

FIGURE 2. Enumeration of positions for 90°, 60°, 45°, and 30° server configurations.

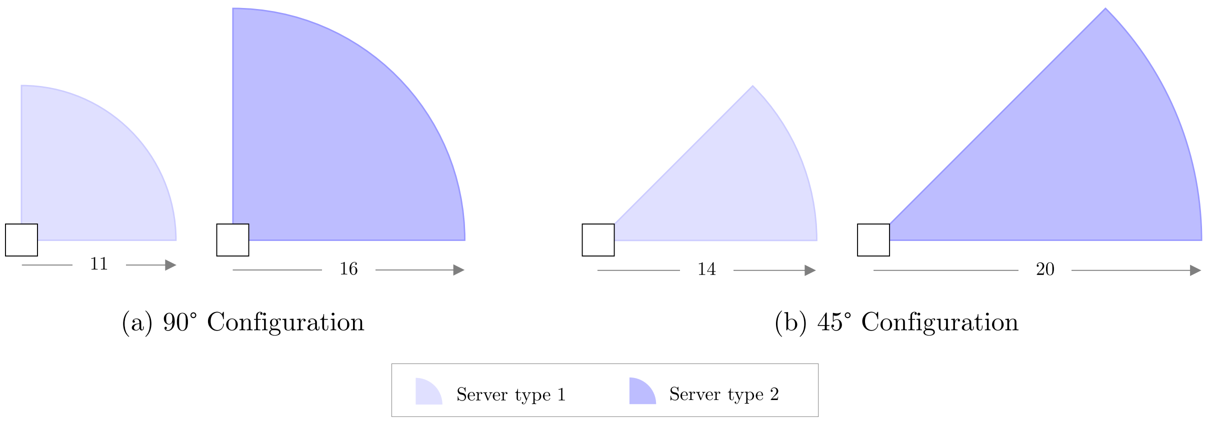

FIGURE 3. Covering angle-distance ratio for the 45° and 90° server configurations.

This formulation selects, if any, at most one server combination per facility location from a set of possible server combinations.

Table 2 provides an additional notation that facilitates this reformulation.

TABLE 2. Additional notation for $\mathbf{ACP}^{\mathrm{MP}}$.

Sets	
R_j:	set of server combinations (t, p, s) for the location j, indexed by a.
Parameters	
c_j^a:	cost of locating a facility at location j with the server combination a.
δ_{ij}^a:	1, if the demand point i is covered from location j with server combination a, 0 otherwise.
Decision variables	
f_j^a:	1, if a facility is located at location j with the server combination a, 0 otherwise.

The master problem $\mathbf{ACP}^{\mathrm{MP}}$ is formulated as follows:

$$\min \quad \sum_{j\in N}\sum_{a\in R_j} c_j^a f_j^a \tag{7}$$

$$\text{s.t.} \quad \sum_{j\in N}\sum_{a\in R_j} \delta_{ij}^a f_j^a \geq 1, \quad \forall i \in M, \tag{8}$$

$$\sum_{a\in R_j} f_j^a \leq 1, \quad \forall j \in N, \tag{9}$$

$$f_j^a \in \{0, 1\}, \quad \forall j \in N. \tag{10}$$

Objective function (7) minimizes the total costs of the network, which involves the cost of locating facilities and installing a combination of servers. Constraints (8) ensure the coverage of all demand points by at least one server combination of the located facilities. Constraints (9) indicate the possibility of selecting at least one server combination per location, and constraints (10) define the domain of the decision variables.

The CG algorithm, inspired by [36], optimally solves the linear relaxation of the ACP model, offering a lower bound for it. This lower bound is attainable by solving the relaxed master problem $\mathbf{ACP}^{RMP}$ using all possible columns. However, the CG algorithm includes only promising columns, thus avoiding a complete enumeration.

First, the reduced master problem, that is, $\mathbf{ACP}^{\mathrm{MP}}$ with only a subset of columns, is solved to obtain an initial solution and dual variables associated with all constraints. Subsequently, in each iteration of the CG algorithm, subproblems $\mathbf{ACP}_j^{\mathrm{SP}}$, based on the dual variables obtained in $\mathbf{ACP}^{RMP}$, identify the most promising column for each location, which are added to the master problem if they have negative reduced cost $\widehat{C}_j^a$. This loop is repeated until no more columns with negative reduced cost are found.

B. THE SUBPROBLEMS

Subproblems aim to generate new feasible columns for each location j, so that the reduced cost is minimized. A new column is accepted if the reduced cost is negative. As mentioned previously, the dual variable values from $\mathbf{ACP}^{\mathrm{RMP}}$ guide the subproblems $\mathbf{ACP}_j^{\mathrm{SP}}$, toward better columns for each location j. The notation for the subproblems is presented in Table 3.

TABLE 3. Additional notation for $\mathbf{ACP}_j^{\mathrm{SP}}$.

Parameters	
θ_i:	Dual value associated to Constraint (8), $\forall i \in M$
λ_j:	Dual value associated to Constraint (9), $\forall j \in N$
Decision variables	
z_{ij}:	1, if demand point i is covered from facility j, 0 otherwise

The subproblems $\mathbf{ACP}_j^{\mathrm{SP}}$ are presented below:

$$\min \quad \widehat{C}_j^a \tag{11}$$

$$\text{s.t.} \quad \sum_{(j,t,p,s)\in Q_i} y_{jt}^{ps} \geq z_{ij}, \quad \forall i \in M, \tag{12}$$

$$\sum_{s\in S} y_{jt}^{ps} \leq 1, \quad \forall t \in T, p \in P_t, \tag{13}$$

$$y_{jt}^{ps} \in \{0,1\}, \quad \forall t \in T, p \in P_t, s \in S_t. \tag{14}$$

$$Z_{ij}^{ps} \in \{0,1\}, \quad \forall i \in M \tag{15}$$

The expression (11) minimizes the reduced costs of facility-server configuration, which means adding a new column for the relaxation master problem. Constraints (12) ensures that at least one server configuration is selected if a facility covers a demand point. Constraints (13) indicate that, at most, one server type with the same configuration and position can be selected. Constraints (14) and (15) specify the domain of the decision variables.

The reduced costs $\widehat{C}_j^a$ are calculated according to the $\mathbf{ACP}^{\mathrm{RMP}}$ as follows:

$$\widehat{C}_j^a = C_j^a - \left(\sum_{i\in M} \theta_i \delta_{ij}^a + \lambda_j\right), \tag{16}$$

However, to propose a new column, a and j are unknown; we represent them in terms of ACP model variables (w_j facility location and y_{jt}^{ps} configuration for a facility). Thus, the reduced cost for a facility with a configuration is:

$$\widehat{C}_j^a = \left(g_j w_j + \sum_{t\in T}\sum_{p\in P_t}\sum_{s\in S_t} c_s^t y_{jt}^{ps}\right) - \left(\sum_{i\in M} \theta_i \delta_{ij}^a + \lambda_j\right), \tag{17}$$

In addition, the parameter δ_{ij}^a of the $\mathbf{ACP}^{\mathrm{MP}}$ depends on the values of the variables z_{ij} of $\mathbf{ACP}^{\mathrm{SP}}(j)$ when a new column a is added. This way, it can communicate with $\mathbf{ACP}^{\mathrm{MP}}$ whether column a covers demand point i from location j. Thus, in the next iteration, it selects at most one column per location and as many columns as needed to cover all demand points. Then, the reduced cost for each facility j:

$$\widehat{C}_j^a = \underbrace{(g_j w_j - \lambda_j)}_{v_1(SP_j)} + \underbrace{\sum_{t\in T}\sum_{p\in P_t}\sum_{s\in S_t} c_s^t y_{jt}^{ps} - \sum_{i\in M}\theta_i z_{ij}}_{v_2(SP_j)} \tag{18}$$

Note that $v_1(SP_j)$ is constant for each $\mathbf{ACP}_j^{\mathrm{SP}}$, that is, facility j is located. If $v_1(SP_j) + v_2(SP_j) < 0$, then the exchange is considered beneficial, and a column is added to the set $\widehat{R}_j$, which is updated by adding the column newly generated by $\mathbf{ACP}_j^{\mathrm{SP}}$. After performing this for all subproblems, the $\mathbf{ACP}^{\mathrm{RMP}}$ has new options to be selected in the next iteration. These steps are repeated until no further columns can be added, meaning that $\mathbf{ACP}_{\mathrm{RMP}}$ has been solved to optimality, thereby providing a lower bound for $\mathbf{ACP}_{\mathrm{MP}}$. To find near-optimal solutions, that is, the upper bounds for $\mathbf{ACP}_{\mathrm{MP}}$, we solve the $\mathbf{ACP}^{\mathrm{MP}}$ with all columns generated. Fig. 4 illustrates the CG algorithm.

A summary of the algorithm steps is as follows:

Step 1: Propose an initial feasible considering a subset of combinations, ensuring that all demand points are covered. Then, it goes to step 2.
Step 2: Solve $\mathbf{ACP}^{\mathrm{RMP}}$, calculating the value of dual variables θ_i and λ_j. Then, it goes to step 3.
Step 3: Solve each $\mathbf{ACP}_j^{\mathrm{SP}}$ considering the value of the dual variables. Then, it goes to step 4.
Step 4: If the solution of subproblem $\mathbf{ACP}_j^{\mathrm{SP}}$ is negative (negative reduced cost), then a new column is included in the list of new variables to add to the master problem. Then, it goes to step 5.
Step 5: If the list of new variables to add to the master problem is not empty, then go to Step 2. Otherwise, it goes to Step 6.
Step 6: Solve $\mathbf{ACP}^{\mathrm{RMP}}$. If the varaibles are integer, then the algorithm finishes, obtaining the optimal solution. Otherwise, it goes to step 7.
Step 7: Solve the $\mathbf{ACP}^{\mathrm{MP}}$ with all the added columns in $\widehat{R}_j$. Then, the algorithm finishes obtaining a feasible solution.

To provide the initial feasible combination for each location, we install a server type that covers the largest area in the minor configuration at all positions in each candidate location. It guarantees the highest coverage from all facilities.

The results of solving the ACP model directly with CPLEX and using the CG algorithm are discussed in Section IV.

IV. COMPUTATIONAL EXPERIENCE

This section describes the computational experiments we performed with 43 sets of demand points, each including four instances obtained by varying the parameters for each set.

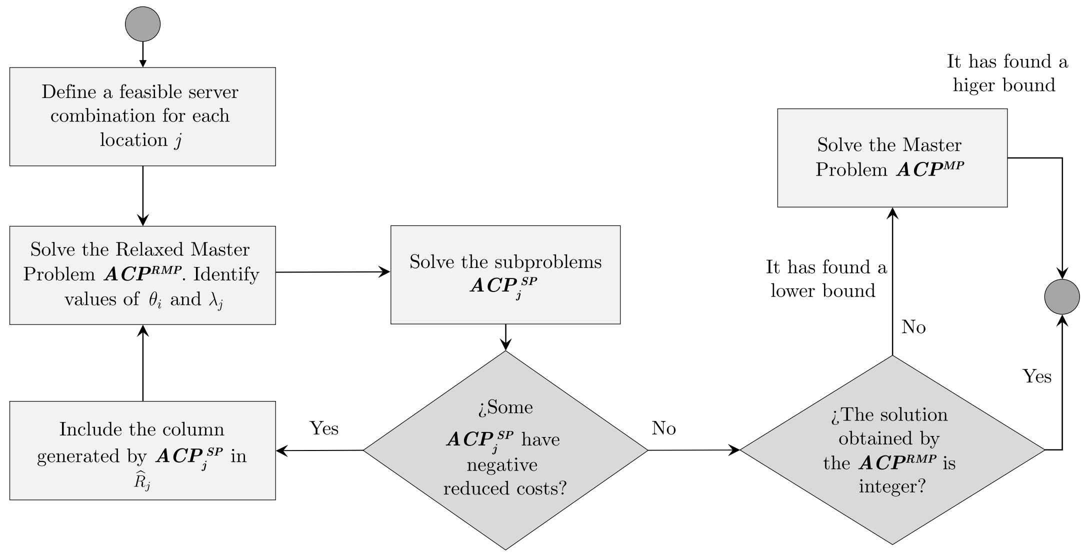

FIGURE 4. Column generation algorithm.

We coded and solved the ACP model and CG algorithm using C++/CPLEX 20.0.1. We ran the tests on an AMD Ryzen 7 5700G CPU with 16.0 GB of RAM and a Windows 11 operating system. We configured CPLEX to use only one thread with a 1-hour time limit for solution computation.

A. TEST INSTANCES GENERATION

An ACP instance comprises demand points, potential facility locations, server types, and server configurations. We systematically produce four instances for each set by leveraging 43 sets of demand points derived from established capacitated vehicle routing (CVRP) instances (Christofides et al.; Fisher; Golden et al.; Christofides and Eilon; Rochat and Taillard). This process involves varying the instance parameters to yield 172 instances. Table 4 outlines the instance generation scheme and providees a comprehensive overview of the parameter variations applied. The number of potential facility locations depends on the number of demand points in the set. In two instances, the number of locations is 20% of the total number of points, whereas in the remaining two instances, it is 50%. We consider either two or four server types for each. Finally, all instances consider four server configurations.

We generate the coordinates of the potential facility locations following a uniform distribution. We link the parameters of this distribution to the coordinates of each set of demand points, ensuring the generated locations are within the minimum and maximum limits along the x- and y-axes. Consequently, the coordinates of the potential locations are consistently distributed randomly within the boundaries defined by the demand point coordinates. This approach guarantees a spatial randomness that aligns with the geographical context of demand points. We chose the rest of the instance parameters to reflect real conditions with broad ranges to evaluate the behavior of the ACP model and CG algorithm in different scenarios, including simple and complex coverage scenarios, ensuring that the instances are representative.

TABLE 4. Instance generation scheme.

Parameter	Values	Description
$\|N\|$	72 - 483	Total number of demand points in the set
$\|M\|$	$0.2\,\|N\|$, $0.5\,\|N\|$	Total number of potential facility locations
$\|S\|$	2, 4	Total number of server types (with different covering distance)
$\|T\|$	4	Total number con server configuration (30°, 45°, 60°, and 90°)

The 172 instances are grouped according to the number of demand points to make the analysis more manageable. The grouping results are listed in Table 5. As shown in the first column, five groups are obtained based on demand points. The second column shows the range, and the third column presents the number of instances in that range. The generated instances are available at https://github.com/fredyandres73/The-Angular-Set-Covering-Problem.

As expected, increasing the number of demand points also increases the time required to solve each instance. Section IV-C shows this trend.

B. BENEFIT OF THE ANGULAR COVERING STRUCTURE

The angular coverage structure adopted by the ACP has advantages over a radial coverage structure, primarily because the area covered to meet the demand is minor. In addition, the radial coverage assumption is only sometimes

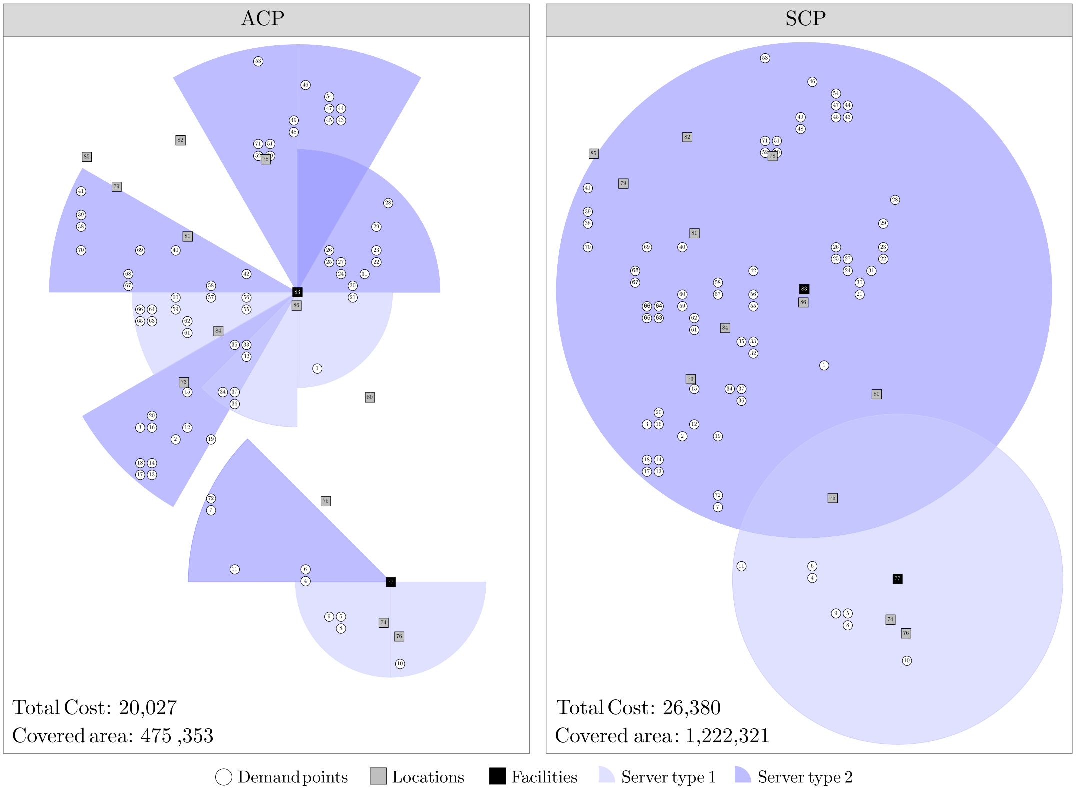

FIGURE 5. Optimal solution for an instance composed of 72 demand points and 14 locations for ACP and SCP.

TABLE 5. Instance grouping based on the number of demand points.

Group	Demand points interval	Instances
1	72-99	28
2	100-199	60
3	200-299	24
4	300-399	36
5	400-483	24

valid in the surveillance context. The same happens if the problem is to locate directional communication antennas to cover demand.

Fig. 5 presents both coverage structures for the same instance, composed of 72 demand points, 14 potential locations, two server types, and four server configurations.

For comparison, we assume that the cost of facility location is the same. In contrast, the server installation cost for the SCP is equivalent to installing all the servers needed to cover the 360° in the ACP; for example, four servers with a 90° configuration are installed in each position. The area covered by a radial coverage structure is more than twice that covered by an angular coverage structure covering the same demand points. Therefore, radial coverage creates unnecessary coverage in spaces with no demand. It is critical to avoid covering spaces with no demand when the cost is associated with establishing coverage. In particular, when analyzing the total implementation cost, there is an increase, for this instance, of more than 20%. In addition, we note that there are areas where double coverage is established, which increases network robustness in the event of a server failure.

C. TEST INSTANCES RESULTS

We compare the performance of the ACP model and CG algorithm for solving all the test instances. If, by reaching the time limit, the ACP model or CG algorithm is still in progress, we can save the best integer solution attained up to that point. This approach ensures a systematic, time-controlled model and algorithm performance evaluation.

The results for each group are presented below. Table 6 presents the results for group 1, which includes 28 instances of 72-99 demand points. The first column displays the dataset and instance, whereas the second, third, and fourth columns show the optimal value, Obj; solution time, T; and optimality gap, OptGap, obtained when solving the ACP model using CPLEX. The fifth and sixth columns display the optimal value, $\mathrm{Obj}_{\mathrm{CG}}$, and the time required to solve the relaxed master problem in the CG algorithm, T_{CG}. If their solution is an integer, solving the $\mathrm{ACP}^{\mathrm{MP}}$ is unnecessary; therefore, $\mathrm{Obj}_{\mathrm{IP}} = \mathrm{Obj}_{\mathrm{CG}}$. The seventh and eighth columns indicate the optimal value, $\mathrm{Obj}_{\mathrm{IP}}$, and time for solving the integer master problem in the column generation algorithm, T_{IP}. The

TABLE 6. ACP model and CG algorithm comparisons for group 1.

#	ACP model			CG algorithm					Gap (%)
	Obj	T (s)	OptGap (%)	Obj_{CG}	T_{CG} (s)	Obj_{IP}	T_{IP} (s)	Gap_{CG} (%)	
1.1	20,027	0.2	0.0	*20,027.0	3.1	20,027	0.00	0.0	0.0
1.2	19,180	0.3	0.0	19,161.0	1.5	19,180	0.03	0.1	0.0
1.3	19,059	9.0	0.0	*19,059.0	9.4	19,059	0.00	0.0	0.0
1.4	18,514	14.4	0.0	18,505.0	1.9	18,530	0.03	0.1	0.1
2.1	29,208	0.1	0.0	*29,208.0	1.1	29,208	0.00	0.0	0.0
2.2	27,793	0.4	0.0	*27,793.0	1.4	27,793	0.00	0.0	0.0
2.3	29,179	1.0	0.0	29,158.0	2.3	29,283	0.05	0.4	0.4
2.4	27,672	1.3	0.0	27,627.5	2.2	27,678	0.03	0.2	0.0
3.1	20,070	0.1	0.0	*20,070.0	0.8	20,070	0.00	0.0	0.0
3.2	18,768	0.2	0.0	18,767.5	0.7	18,768	0.02	0.0	0.0
3.3	19,847	0.4	0.0	19,770.5	2.3	19,847	0.03	0.4	0.0
3.4	18,650	2.1	0.0	*18,650.0	1.5	18,650	0.00	0.0	0.0
4.1	29,750	0.1	0.0	*29,750.0	2.3	29,750	0.00	0.0	0.0
4.2	21,120	0.2	0.0	*21,120.0	3.6	21,120	0.00	0.0	0.0
4.3	28,934	2.4	0.0	25,932.0	4.2	29,466	0.11	12.0	1.8
4.4	20,309	4.5	0.0	20,290.8	4.5	20,338	0.05	0.2	0.1
5.1	26,602	0.0	0.0	*26,602.0	0.3	26,602	0.00	0.0	0.0
5.2	18,538	0.2	0.0	*18,538.0	0.7	18,538	0.00	0.0	0.0
5.3	20,034	0.3	0.0	*20,034.0	1.7	20,034	0.00	0.0	0.0
5.4	18,538	1.6	0.0	*18,538.0	1.2	18,538	0.00	0.0	0.0
6.1	40,898	0.6	0.0	40,805.8	3.8	41,215	0.15	1.0	0.8
6.2	23,771	0.3	0.0	*23,771.0	5.6	23,771	0.00	0.0	0.0
6.3	34,058	47.5	0.0	*34,058.0	11.4	34,058	0.00	0.0	0.0
6.4	23,771	7.3	0.0	23,612.5	14.6	23,935	0.14	1.3	0.7
7.1	40,339	0.3	0.0	40,213.8	4.8	40,341	0.09	0.3	0.0
7.2	30,404	0.6	0.0	*30,404.0	5.9	30,404	0.00	0.0	0.0
7.3	39,634	41.3	0.0	39,582.5	5.1	39,659	0.07	0.2	0.1
7.4	25,132	2.4	0.0	*25,132.0	16.0	25,132	0.00	0.0	0.0
Avg	-	5.0	0.0	-	4.1	-	0.03	0.6	0.1

* The optimal solution of the ACP^{RMP} is integer. ** The CG algorithm has reached the time limit.

ninth column indicates the gap between the optimal value obtained by the relaxed master problem and the integer master problem, Gap_{CG}, which is computed as:

$$Gap_{CG} = \frac{Obj_{IP} - Obj_{CG}}{Obj_{IP}} \tag{19}$$

Finally, the tenth column shows the gap between the best solution found by the ACP model, Obj, and the best integer solution found by the CG algorithm, Obj_{IP}, calculated as follows:

$$Gap = \frac{Obj_{IP} - Obj}{Obj_{IP}} \tag{20}$$

A positive value indicates that, on average, the ACP model yields better solutions. Conversely, a negative value means that the CG algorithm, on average, outperforms the ACP model by providing better solutions compared to the best integer solution.

The ACP model finds optimal solutions for all group instances within the time limit, whereas the CG algorithm finds optimal solutions in 19 instances. The performance gap between the ACP model and the CG algorithm is remarkably small, with the most significant difference not exceeding 2%. There are differences in the solution times. Depending on the instance, either the ACP model or CG algorithm is faster. However, the average CG time of 4.1 seconds is less than that of the ACP model solution time of 5 seconds.

Table 7 presents the results for group 2, which consisted of 60 instances with 100-199 demand points.

The ACP model finds optimal solutions in 57 instances within the given time limit, whereas the CG algorithm finds optimal solutions in 40 instances. For the remaining instances, the CG algorithm demonstrates its competitiveness, with a performance gap of less than 3% compared to the solution obtained by the ACP model. It is possible to find instances in which the solution obtained through the CG algorithm is better than that of the ACP model by 5%. Regarding the solution times, while some instances solved through the ACP model reached the time limit, the longest CG algorithm time is 132 seconds (instance 22.4). On average,

TABLE 7. ACP model and CG algorithm comparisons for group 2.

	ACP model			CG algorithm					
#	Obj	T (s)	OptGap (%)	Obj_{CG}	T_{CG} (s)	Obj_{IP}	T_{IP} (s)	Gap_{CG} (%)	Gap (%)
8.1	52,680	0.4	0.0	*52,680.0	6.0	52,680	0.00	0.0	0.0
8.2	34,787	1.1	0.0	*34,787.0	8.7	34,787	0.00	0.0	0.0
8.3	44,055	10.5	0.0	44,022.5	19.3	44,068	0.11	0.1	0.0
8.4	34,106	13.3	0.0	33,952.2	23.8	34,132	0.23	0.5	0.1
9.1	37,876	0.0	0.0	*37,876.0	1.2	37,876	0.00	0.0	0.0
9.2	22,930	0.3	0.0	*22,930.0	4.2	22,930	0.00	0.0	0.0
9.3	30,867	1.1	0.0	*30,867.0	5.0	30,867	0.00	0.0	0.0
9.4	22,170	20.3	0.0	*22,170.0	6.7	22,170	0.00	0.0	0.0
10.1	30,648	0.2	0.0	*30,648.0	3.6	30,648	0.00	0.0	0.0
10.2	28,294	0.7	0.0	25,172.7	3.0	29,049	0.04	13.3	2.6
10.3	30,613	5.5	0.0	*30,613.0	12.5	30,613	0.00	0.0	0.0
10.4	21,754	5.9	0.0	*21,754.0	21.0	21,754	0.00	0.0	0.0
11.1	39,061	0.2	0.0	*39,061.0	2.3	39,061	0.00	0.0	0.0
11.2	23,018	0.1	0.0	*23,018.0	5.1	23,018	0.00	0.0	0.0
11.3	30,587	1.8	0.0	30,495.0	6.5	30,653	0.08	0.5	0.2
11.4	22,036	2.7	0.0	22,003.7	13.1	22,058	0.08	0.2	0.1
12.1	28,825	0.3	0.0	*28,825.0	1.2	28,825	0.00	0.0	0.0
12.2	19,716	0.1	0.0	*19,716.0	3.3	19,716	0.00	0.0	0.0
12.3	21,819	8.9	0.0	*21,819.0	4.0	21,819	0.00	0.0	0.0
12.4	19,682	28.7	0.0	*19,682.0	5.9	19,682	0.00	0.0	0.0
13.1	38,841	0.3	0.0	*38,841.0	5.7	38,841	0.00	0.0	0.0
13.2	22,438	0.3	0.0	*22,438.0	7.9	22,438	0.00	0.0	0.0
13.3	32,161	14.7	0.0	*32,161.0	16.6	32,161	0.00	0.0	0.0
13.4	22,438	3.8	0.0	*22,438.0	23.4	22,438	0.00	0.0	0.0
14.1	9,384	0.0	0.0	*9,384.0	0.3	9,384	0.00	0.0	0.0
14.2	9,052	0.0	0.0	*9,052.0	0.4	9,052	0.00	0.0	0.0
14.3	9,384	0.0	0.0	*9,384.0	0.5	9,384	0.00	0.0	0.0
14.4	9,052	0.1	0.0	*9,052.0	1.1	9,052	0.00	0.0	0.0
15.1	39,440	0.1	0.0	*39,440.0	1.3	39,440	0.00	0.0	0.0
15.2	37,427	0.5	0.0	34,207.0	2.3	37,546	0.04	8.9	0.3
15.3	39,440	1.1	0.0	*39,440.0	3.1	39,440	0.00	0.0	0.0
15.4	30,221	39.2	0.0	*30,221.0	4.6	30,221	0.00	0.0	0.0
16.1	30,218	0.2	0.0	*30,218.0	3.3	30,218	0.00	0.0	0.0
16.2	21,505	0.7	0.0	*21,505.0	3.5	21,505	0.00	0.0	0.0
16.3	29,353	17.7	0.0	29,254.5	6.2	29,353	0.06	0.3	0.0
16.4	20,284	83.9	0.0	*20,284.0	5.3	20,284	0.00	0.0	0.0
17.1	71,801	0.6	0.0	68,531.0	12.9	72,886	0.63	6.0	1.5
17.2	53,085	1.6	0.0	52,814.6	9.0	53,586	0.46	1.4	0.9
17.3	63,957	3,600.0	5.3	63,867.0	38.2	63,957	0.19	0.1	0.0
17.4	45,410	3,164.1	0.0	45,005.8	53.1	45,928	1.33	2.0	1.1
18.1	38,353	0.4	0.0	35,520.0	4.4	38,640	0.04	8.1	0.7
18.2	22,022	0.3	0.0	*22,022.0	4.5	22,022	0.00	0.0	0.0
18.3	37,703	89.8	0.0	34,745.0	13.3	38,222	0.08	9.1	1.4
18.4	22,022	4.1	0.0	*22,022.0	9.8	22,022	0.00	0.0	0.0
19.1	48,324	0.1	0.0	*48,324.0	2.7	48,324	0.00	0.0	0.0
19.2	37,222	4.0	0.0	32,833.0	4.4	38,123	0.04	13.9	2.4
19.3	39,932	16.0	0.0	*39,932.0	4.2	39,932	0.00	0.0	0.0
19.4	30,267	309.1	0.0	30,200.8	8.0	30,280	0.07	0.3	0.0
20.1	45,905	0.5	0.0	*45,905.0	4.1	45,905	0.00	0.0	0.0
20.2	36,720	1.3	0.0	33,307.3	4.4	37,686	0.06	11.6	2.6
20.3	40,805	103.9	0.0	*40,805.0	4.6	40,805	0.00	0.0	0.0

TABLE 7. *(Continued.)* **ACP model and CG algorithm comparisons for group 2.**

20.4	30,770	76.2	0.0	30,599.5	7.6	31,357	0.12	2.4	1.9
21.1	45,982	0.1	0.0	*45,982.0	4.3	45,982	0.00	0.0	0.0
21.2	30,436	0.6	0.0	*30,436.0	13.5	30,436	0.00	0.0	0.0
21.3	38,971	2.2	0.0	*38,971.0	12.5	38,971	0.00	0.0	0.0
21.4	30,212	72.9	0.0	*30,212.0	28.2	30,212	0.00	0.0	0.0
22.1	92,062	47.9	0.0	87,596.1	25.5	93,325	5.31	6.1	1.4
22.2	57,812	37.2	0.0	57,556.9	20.0	58,445	1.09	1.5	1.1
22.3	83,549	3,600.0	12.6	78,380.2	60.9	79,104	0.71	0.9	-5.6
22.4	58,140	3,600.0	25.4	54,551.5	71.6	59,028	59.77	7.6	1.5
Avg	-	250.0	0.7	-	11.1	-	1.18	1.6	0.2

* The optimal solution of the ACP^{RMP} is integer. ** The CG algorithm has reached the time limit.

TABLE 8. **ACP model and CG algorithm comparisons for group 3.**

	ACP model			CG algorithm					
#	Obj	T (s)	OptGap (%)	Obj_{CG}	T_{CG} (s)	Obj_{IP}	T_{IP} (s)	Gap_{CG} (%)	Gap (%)
23.1	93,530	40	0.0	93,042.7	14.7	94,638	1.35	1.7	1.2
23.2	58,618	102	0.0	58,287.1	23.7	59,364	1.64	1.8	1.3
23.3	85,342	3,600	16.2	80,802.8	30.2	86,993	29.89	7.1	1.9
23.4	57,272	3,600	24.7	51,687.3	76.1	51,862	0.36	0.3	-10.4
24.1	108,337	9	0.0	108,136.0	32.2	109,406	1.84	1.2	1.0
24.2	72,487	2,011	0.0	71,217.9	20.8	74,213	21.24	4.0	2.3
24.3	99,826	3,600	23.4	92,821.7	49.0	94,647	2.87	1.9	-5.5
24.4	70,831	3,600	31.7	66,837.0	29.1	74,401	183.15	10.2	4.8
25.1	82,649	2	0.0	82,046.7	28.9	83,165	0.55	1.3	0.6
25.2	54,551	13	0.0	54,529.2	38.9	54,592	0.24	0.1	0.1
25.3	73,816	3,600	8.7	*73,816.0	71.8	73,816	0.00	0.0	0.0
25.4	47,174	3,600	3.5	47,058.6	92.9	47,315	0.73	0.5	0.3
26.1	147,411	9	0.0	147,109.0	23.4	152,238	43.67	3.4	3.2
26.2	94,976	217	0.0	94,281.6	39.6	96,607	5.87	2.4	1.7
26.3	133,921	3,600	12.3	130,587.0	41.3	137,655	266.66	5.1	2.7
26.4	94,476	3,600	22.7	89,400.2	49.1	89,612	0.69	0.2	-5.4
27.1	85,361	7	0.0	81,770.1	28.2	87,281	8.04	6.3	2.2
27.2	50,322	18	0.0	50,252.0	42.5	50,520	0.28	0.5	0.4
27.3	77,888	3,600	25.5	74,700.5	46.3	78,680	48.52	5.1	1.0
27.4	49,069	3,600	5.5	48,891.8	114.1	49,152	1.30	0.5	0.2
28.1	124,528	156	0.0	120,612.0	44.4	127,290	211.12	5.2	2.2
28.2	81,601	841	0.0	81,125.4	49.1	81,914	2.95	1.0	0.4
28.3	110,828	3,600	21.1	103,480.0	136.8	113,246	464.93	8.6	2.1
28.4	80,707	3,600	31.4	74,328.4	47.0	83,021	401.45	10.5	2.8
Avg	-	1,943	9.5	-	48.8	-	70.81	3.3	0.5

* The optimal solution of the ACP^{RMP} is integer. ** The CG algorithm has reached the time limit.

the ACP model takes 250.0 seconds to solve, whereas the CG algorithm finds solutions with objectives 0.2% above in 12.3 seconds.

Table 8 presents the results for group 3, which consists of 24 instances of 200-299 demand points.

The ACP model determines the optimal solutions in 12 instances. However, the CG algorithm does not find optimal solutions for any instance. Nevertheless, their gap compared to the model is not higher than 3.2%. For instances in which the ACP model does not find optimal solutions, the CG algorithm takes advantage. It improves the solution quality obtained by the ACP model by up to 10% (instance 23.4). While the ACP model reaches the time limit in 12 out of 20 instances, the CG algorithm finds its best integer solution in 602 seconds at most (instance 28.3). On average, the ACP model takes 1,943 seconds to solve, whereas the CG algorithm finds solutions with objectives 0.5% above in 119.6 seconds.

Table 9 presents the results for group 4, which consist of 36 instances of demand points between 300-399.

TABLE 9. ACP model and CG algorithm comparisons for group 4.

#	ACP model			CG algorithm					Gap (%)
	Obj	T (s)	OptGap (%)	Obj_{CG}	T_{CG} (s)	Obj_{IP}	T_{IP} (s)	Gap_{CG} (%)	
29.1	93,568	119	0.0	89,408.0	47.0	95,441	183.20	6.3	2.0
29.2	65,277	204	0.0	64,342.6	41.2	66,118	1.56	2.7	1.3
29.3	85,402	3,600	20.6	84,935.4	75.3	85,881	4.95	1.1	0.6
29.4	63,339	3,600	28.3	60,562.1	60.1	63,597	3.11	4.8	0.4
30.1	175,190	12	0.0	174,854.0	70.6	176,329	3.26	0.8	0.6
30.2	117,001	1,277	0.0	113,922.0	85.0	119,696	174.71	4.8	2.3
30.3	172,433	3,600	22.0	161,167.0	82.0	169,413	3,413.32	4.9	-1.8
30.4	116,582	3,600	27.7	105,709.0	89.6	114,056	531.22	7.3	-2.2
31.1	133,802	84	0.0	133,601.0	68.8	134,801	2.77	0.9	0.7
31.2	93,769	3,600	0.8	92,554.3	48.6	95,573	18.10	3.2	1.9
31.3	138,425	3,600	27.7	124,685.0	57.1	130,819	216.47	4.7	-5.8
31.4	101,707	3,600	36.8	85,882.4	73.5	90,761	507.36	5.4	-12.1
32.1	98,368	17	0.0	97,897.7	57.2	99,536	4.54	1.6	1.2
32.2	62,699	474	0.0	62,799.2	75.9	62,887	0.64	0.1	0.3
32.3	97,484	3,600	19.6	93,916.5	145.9	99,711	432.23	5.8	2.2
32.4	62,189	3,600	15.3	60,423.7	86.3	63,663	9.68	5.1	2.3
33.1	98,317	6	0.0	98,374.0	65.6	98,374	0.24	0.0	0.1
33.2	75,955	2,363	0.0	75,717.6	84.9	76,504	3.15	1.0	0.7
33.3	97,963	3,600	20.8	96,066.3	79.9	100,051	56.01	4.0	2.1
33.4	21,944	303	0.0	*21,944.0	64.4	21,944	0.00	0.0	0.0
34.1	153,296	554	0.0	149,514.0	49.3	156,617	407.27	4.5	2.1
34.2	97,710	3,600	8.5	94,671.2	72.2	100,270	131.81	5.6	2.6
34.3	150,388	3,600	30.0	134,057.0	75.9	142,678	1,003.75	6.0	-5.4
34.4	103,218	3,600	29.2	91,659.2	115.9	94,460	29.83	3.0	-9.3
35.1	110,848	19	0.0	110,336.0	56.1	111,463	0.60	1.0	0.6
35.2	68,083	92	0.0	67,705.3	79.5	68,487	1.20	1.1	0.6
35.3	94,805	3,600	8.6	94,209.2	168.1	95,427	5.58	1.3	0.7
35.4	66,675	3,600	16.5	66,338.1	142.2	66,644	0.88	0.5	0.0
36.1	215,812	590	0.0	214,770.0	87.6	219,242	403.53	2.0	1.6
36.2	139,566	1,658	0.0	138,760.0	116.9	144,940	699.73	4.3	3.7
36.3	207,184	3,600	20.8	191,143.0	111.0	199,590	1,900.17	4.2	-3.8
36.4	143,393	3,600	27.4	128,615.0	152.1	135,259	638.12	4.9	-6.0
37.1	121,139	204	0.0	120,235.0	73.1	122,977	21.60	2.2	1.5
37.2	82,164	3,600	7.9	78,048.6	61.9	84,686	149.28	7.8	3.0
37.3	120,336	3,600	25.0	109,601.0	97.3	116,523	619.61	5.9	-3.3
37.4	82,496	3,600	21.3	72,978.2	99.0	77,283	93.09	5.6	-6.7
Avg	-	2,222	11.5	-	83.8	-	324.24	3.5	-0.6

* The optimal solution of the ACP^{RMP} is integer. ** The CG algorithm has reached the time limit.

The ACP model finds optimal solutions in 16 instances within the time limit, whereas more than 20 instances have optimality gaps that reach more than 30% in some instances. The CG algorithm matches one optimal solution found by the ACP model. It also provides better solutions than the ACP model in 10 instances, improving the quality of the solutions by up to 13% (instance 31.4). Regarding solution times, the ACP model reaches the solution time limit in more than half of the instances. In contrast, the average solution time of the CG algorithm is close to 408 seconds. On average, the ACP model takes 2,222 seconds to solve, whereas the CG algorithm finds solutions in 408 seconds, improving their quality by 0.6%.

Table 10 presents results for 24 instances with 400-483 demand points in group 5.

The ACP model finds optimal solutions in only four instances within the time limit, and the other 20 instances have an optimality gap of 42%. The CG algorithm consistently outperforms the ACP model, finding better solutions in 11 instances. This improvement in solution quality is significant, with the CG algorithm enhancing the quality of the solutions obtained by up to 30% (instance 42.4). Regarding the solution times, the CG algorithm demonstrates

TABLE 10. ACP model and CG algorithm comparisons for group 5.

#	ACP model			CG algorithm					Gap (%)
	Obj	Time (s)	OptGap (%)	Obj_{CG}	$Time_{CG}$ (s)	Obj_{IP}	$Time_{IP}$ (s)	Gap_{CG} (%)	
38.1	166,835	51	0.0	166,736.0	77.7	167,370	0.95	0.4	0.3
38.2	110,118	3,600	6.8	106,742.0	102.6	119,908	**3,497.45	11.0	8.2
38.3	177,603	3,600	32.4	149,052.0	100.8	162,438	**3,499.24	8.2	-9.3
38.4	108,392	3,600	22.9	100,913.0	249.4	110,678	2,154.78	8.8	2.1
39.1	125,646	542	0.0	124,488.0	64.5	128,365	108.57	3.0	2.1
39.2	78,985	3,600	8.4	78,526.2	137.9	79,177	5.04	0.8	0.2
39.3	115,989	3,600	27.0	109,655.0	160.6	111,768	28.30	1.9	-3.8
39.4	78,472	3,600	17.2	77,394.1	286.9	78,236	12.42	1.1	-0.3
40.1	191,795	3,600	1.8	186,936.0	65.0	199,364	**3,534.98	6.2	3.8
40.2	119,610	3,600	15.9	113,502.0	112.5	124,181	2,202.23	8.6	3.7
40.3	183,177	3,600	29.9	164,177.0	128.3	173,690	**3,471.68	5.5	-5.5
40.4	135,452	3,600	30.6	108,521.0	180.6	114,771	264.69	5.4	-18.0
41.1	268,017	85	0.0	266,976.0	157.1	270,262	128.21	1.2	0.8
41.2	170,538	3,600	11.0	163,958.0	152.7	179,519	**3,447.34	8.7	5.0
41.3	252,171	3,600	28.0	230,842.0	171.9	242,543	**3,428.15	4.8	-4.0
41.4	184,040	3,600	34.2	152,110.0	169.1	165,265	**3,430.93	8.0	-11.4
42.1	193,504	387	0.0	189,702.0	106.3	198,472	733.22	4.4	2.5
42.2	132,113	3,600	10.3	128,001.0	139.0	135,644	783.21	5.6	2.6
42.3	202,547	3,600	32.8	175,598.0	136.5	187,660	**3,463.53	6.4	-7.9
42.4	159,922	3,600	42.2	117,011.0	222.6	122,271	271.58	4.3	-30.8
43.1	149,895	3,600	2.1	142,015.0	108.5	154,155	**3,491.50	7.9	2.8
43.2	89,316	3,600	6.5	88,905.3	181.6	90,249	8.85	1.5	1.0
43.3	143,532	3,600	30.1	131,939.0	256.0	139,678	2,671.27	5.5	-2.8
43.4	95,905	3,600	21.2	86,190.9	204.0	89,331	44.68	3.5	-7.4
Avg	-	3,044	17.1	-	153.0	-	1,695.12	5.1	-2.7

* The optimal solution of the ACP^{RMP} is integer. ** The CG algorithm has reached the time limit.

its efficiency by reaching the time limit in only nine instances, compared to the ACP model in 20 instances. On average, the CG algorithm significantly outperforms the ACP model, finding solutions in 1,848 seconds compared to the ACP model with 3,044 seconds. This represents a 2.7% improvement in solution quality. As seen in the results per group, the CG algorithm takes advantage of the ACP model when the size of the instances to be solved increases. Table 11 provides an average comparison between the ACP model and CG algorithm for all the instances. The first and second columns denote the group and quantity of instances, respectively. Subsequently, the third and fourth columns present the average solution time and average optimality gap for the ACP model solved using CPLEX. The fifth and sixth columns show the average solution time and the average gap of the CG algorithm compared with the best integer solution obtained by the ACP model. As shown in Tables 6 to 10, the solution provided by the CG algorithm is better if it is negative.

On average, no significant differences are observed in group 1. However, as the size of the instances to be solved increases, the solution quality and computational time differences increase. The CG algorithm achieves reductions of 93%, 89%, 79%, and 37% in the solution time compared to the ACP model for groups 2, 3, 4, and 5, respectively. Regarding solution quality, the average gap between the solutions obtained by the CG algorithm and ACP model is less than 1% in groups 1, 2, and 3. This suggests that, on average, the quality of the solutions obtained by the CG algorithm is nearly equivalent to that obtained by the ACP model. For groups 4 and 5, the CG algorithm finds better solutions than those obtained by the ACP model. The computation times are significantly lower, particularly for the most challenging groups. In addition, better-quality solutions are possible, which is promising for solving even larger instances, such as real cases.

We also address real case-solving instances for locating surveillance security cameras in Valparaíso, Chile. Those results are discussed in section V.

V. THE CASE OF VALPARAÍSO, CHILE

Valparaíso is a city port in Chile, the capital of the province and the Region of Valparaíso. It is the seat of the Chilean legislature and home of the Chilean Navy. Valparaíso has a population of approximately 300,000 and is considered a world heritage site. It is known for its steep funiculars and colorful houses on the hills. In 2003, the historic quarter of Valparaíso was declared a UNESCO World Heritage Site.

TABLE 11. ACP model and CG algorithm comparisons per group.

		ACP model		CG algorithm		
Group	Instances	Avg Time (s)	Avg OptGap (%)	Avg Time (s)	Avg Gap_{CG} (%)	Avg Gap (%)
Group 1	28	5.0	0.00	4.1	0.6	0.1
Group 2	60	250.0	0.72	12.2	1.6	0.2
Group 3	24	1,942.7	9.45	119.6	3.3	0.5
Group 4	36	2,221.6	11.53	408.0	3.5	-0.6
Group 5	24	3,044.4	17.14	1,848.1	5.1	-2.7
Avg	-	1,248.9	6.4	364.89	2.5	-0.3

Our focus is on the critical task of locating cameras to establish surveillance coverage across the city. This implies defining geographical demand points that require effective coverage. Our goal is to improve the placement of cameras to enhance the effectiveness of the surveillance systems. We will consider the installation cost, coverage distance, and viewing angle to optimize the placement of the cameras. This will contribute to the safety and security of the city.

A. THE CASE OF VALPARAÍSO INSTANCES GENERATION

A unique set of 1,000 demand points has been created throughout the city. Each demand point represents a geographic location that needs to be covered by at least one surveillance camera. Fig. 6 shows the demand points distribution in Valparaíso.

The potential locations of the facility are part of the demand point set. If a facility is placed at a demand point, it must be covered by a camera belonging to the other facility. The selection of this potential facility subset is random, ensuring that all demand points could be covered from at least one facility location. We use the scheme presented in Table 4 to create instances that incorporate cases where all demand points are potential locations.

Table 12 details the 12 instances for Valparaíso city. The first column details the instance number identification, while the second column indicates the number of demand points, which is the same for all. The third column shows the number of potential facility locations available. The fourth column lists the number of available server types. The fifth column displays the number of server configurations available, considering 30°, 45°, 60° and 90° for the case with four configurations, including 120° and 180° for the case with six configurations.

The objective is to cover all demand points while minimizing the cost of locating facilities and installing servers. All costs are in dollars. As each server type covers a fixed area, decreasing the covering angle increases the covering distance, maintaining the area covered. Therefore, each server type has a fixed installation cost at a larger covering angle (and shorter distance), which increases as the angle decreases (allowing it to reach larger distances). The cost increase is proportional to the increase in coverage distance.

TABLE 12. Description of Valparaíso instances.

Instance	Demand points	Candidate Locations	Server types	Server configurations
Valp-1	1,000	200	2	4
Valp-2	1,000	200	2	6
Valp-3	1,000	200	4	4
Valp-4	1,000	200	4	6
Valp-5	1,000	500	2	4
Valp-6	1,000	500	2	6
Valp-7	1,000	500	4	4
Valp-8	1,000	500	4	6
Valp-9	1,000	1,000	2	4
Valp-10	1,000	1,000	2	6
Valp-11	1,000	1,000	4	4
Valp-12	1,000	1,000	4	6

B. VALPARAÍSO RESULTS

We compare the solutions of the Valparaiso case by directly solving each instance with the ACP model and the CG algorithm. The results for the case study of Valparaíso are presented in Table 13 considering the 1-hour time limit. The information is presented using the same notation in Section IV-C. However, the first column now reports the number of variables for the ACP model (Vars) and the number of columns generated for the CG algorithm (Cols). This measure helps to understand the problem size in terms of the number of variables. Remember that the generated columns in the CG algorithm are variables for the $\mathbf{ACP}^{\mathrm{RMP}}$.

The average OptGap reaching is 46% for the ACP model, while for the CG, it is 20.3%, considering the time limit. Note that T_{CG} is solved for an average of 798 seconds, using the remaining time of 3,600 seconds to obtain a feasible solution through ACP^{MP}. Comparing Gap_{CG} with OptGap across all instances reveals consistently lower values favoring the CG algorithm.

The results reveal distinct trends. Considering the 200 candidate locations, instances from Valp-1 to Valp-4 exhibit no significant differences. On average, the CG algorithm

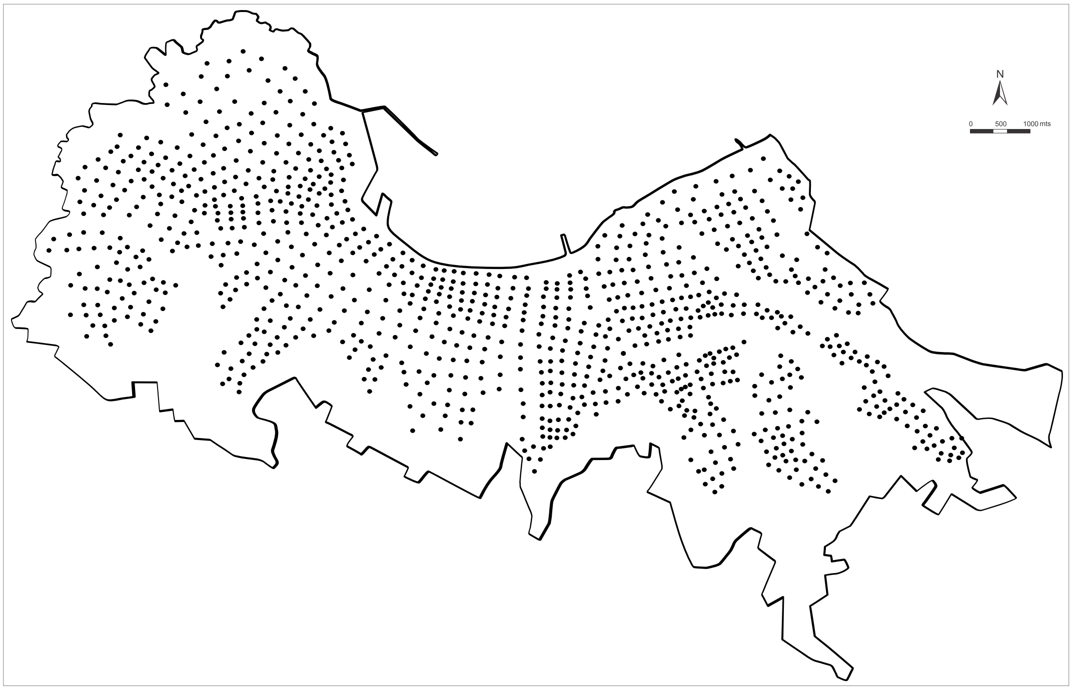

FIGURE 6. Demand points of Valparaíso city.

TABLE 13. ACP model and CG algorithm comparisons for Valparaíso instances.

	ACP model				CG algorithm						
#	Vars	Obj	Time (s)	OptGap (%)	Cols	Obj_{CG}	$Time_{CG}$ (s)	Obj_{IP}	$Time_{IP}$ (s)	Gap_{CG} (%)	Gap (%)
Valp-1	12,200	536,970	3,600	14.4	5,080	474,066.9	354.6	542,828	3,245	12.7	1.1
Valp-2	14,200	542,383	3,600	20.3	5,179	472,680.0	312.5	546,916	3,288	13.6	0.8
Valp-3	24,200	381,426	3,600	34.0	5,068	298,570.0	320.3	367,800	3,280	18.8	-3.7
Valp-4	28,200	378,590	3,600	36.2	4,955	297,892.0	356.9	374,367	3,243	20.4	-1.1
Valp-5	30,500	580,494	3,600	44.0	6,762	415,564.3	543.0	520,950	3,057	20.2	-11.4
Valp-6	35,500	590,214	3,600	50.2	6,958	414,020.0	661.9	523,147	2,938	20.9	-12.8
Valp-7	60,500	393,569	3,600	46.7	7,353	281,724.0	851.0	364,358	2,749	22.7	-8.0
Valp-8	70,500	437,533	3,600	55.4	7,291	280,929.0	838.8	359,727	2,761	21.9	-21.6
Valp-9	61,000	625,763	3,600	56.3	8,133	396,877.0	1,117.3	505,168	2,483	21.4	-23.9
Valp-10	71,000	842,232	3,600	69.2	8,328	395,851.0	1,140.1	503,389	2,460	21.4	-67.3
Valp-11	121,000	486,528	3,600	59.0	9,681	273,591.0	1,492.5	361,172	2,108	24.2	-34.7
Valp-12	141,000	522,290	3,600	66.5	9,977	272,815.0	1,589.2	364,178	2,011	25.1	-43.4
Avg	-	-	3,600	46.0	-	-	798.2	-	2,802	20.3	-18.8

enhances the solution quality obtained by the ACP model by 0.7%, with improvements of 3.7%. More significant differences in solution quality are observed for instances from Valp-5 to Valp-8, with 500 candidate locations, where the CG algorithm achieves an average improvement of 13.5% of the gap. Significant differences are observed for instances Valp-9 to Valp-12, where all demand points are candidate locations. The CG algorithm improves the gap by an average of 42.3%. The result obtained in one of the most challenging instances (Valp-10) shows that the CG algorithm has asignificant advantage over the ACP model, enhancing the best solution by 67.3%.

As shown in the table 13, for all instances, the number of variables in the ACP model is considerably higher than the number of columns generated by the CG algorithm. This significant reduction in the number of variables leads to smaller and easier models to solve. Consequently, this reduction translates into faster solution times, improving solution quality within the time limit. This demonstrates the efficiency and effectiveness of the CG approach in optimizing large-scale problems by decomposing the ACP model and improving the computational performance.

Fig. 7 presents the map of Valparaíso with the obtained solution, for instance Valp-10, where the CG algorithm

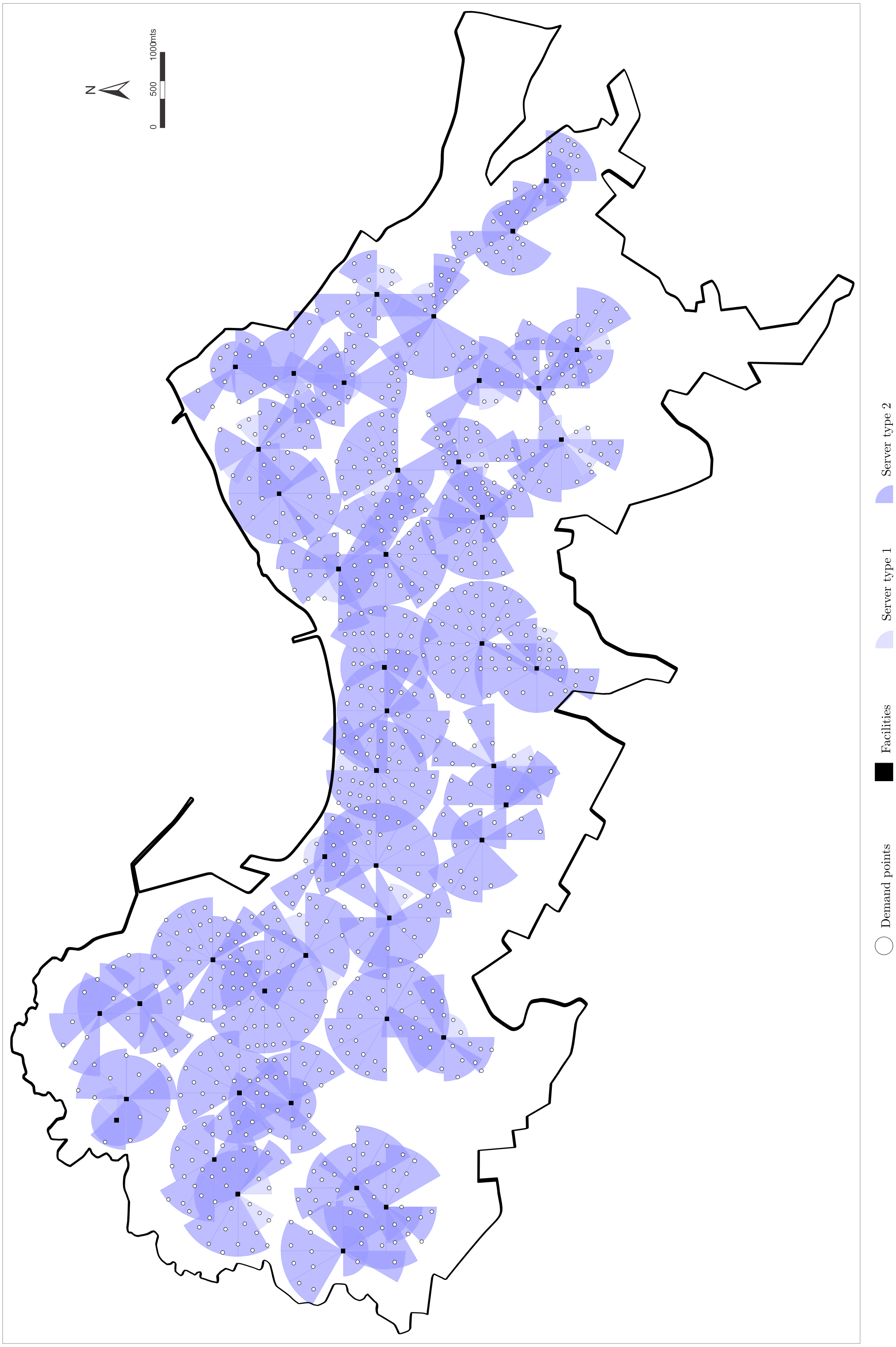

FIGURE 7. Best solution found by CG algorithm for Valp-10 instance (67.3% improvement compared to ACP model).

obtains the best performance compared to the ACP model. Valp-10 comprises 1,000 demand points, 1,000 locations for facilities, two server types, and six server configurations.

VI. CONCLUSION AND FUTURE WORK

We present a formulation for the ACP, an extension of the SCP, which adopts an angular covering structure instead of a radial covering of 360° as in the original SCP. This problem addresses the location of facilities and installation of servers to establish an angular covering that considers different configurations (angles) and server types. As shown in Fig. 5, this angular coverage structure reduces the total location costs and considerably decreases the area covered.

A column generation (CG) algorithm is proposed to solve the ACP. The CG algorithm reduces solution times to obtain optimal solutions or improves the quality of the best integer solution obtained by the ACP model up to the time limit, particularly for larger instances.

We solve 172 instances for the ACP, generated from 43 sets of demand points derived from the established CVRP instances. Tables 6 and 7 show that the CG algorithm can obtain optimal solutions, improving the solution times compared with directly solving the ACP model with CPLEX. Tables 8, 9, and 9 indicate that as the size of the instances increases, the CG algorithm takes advantage of finding better solutions than the ACP model. The improvement in the quality of the best solutions obtained by the model reaches up to 30%.

We implement and solve real-size instances: the case of locating surveillance cameras in Valparaiso. We elaborate instances varying several parameters, such as the number of candidate facilities, server configurations, and server types based on 1,000 demand points in the city. As shown in Table 13, the CG algorithm performs best in solving these instances, improving the solution quality by up to 67% compared with the best solution obtained by the ACP model.

Based on evaluating the performance of the CG algorithm across all the instances solved, we identify opportunities to enhance the solution process as the instance size increases. Our results indicate that the CG approach remains scalable and can obtain better results for even larger instances. Several algorithmic enhancements can improve its efficiency, including improving the initial feasible solution, solving subproblems heuristically instead of optimally with a commercial solver to reduce computational times, and incorporating a column administration process to delete or combine columns. These improvements optimize the algorithm performance and scalability.

There are two possible avenues for future research. First, there is a need to enhance the quality of the obtained solutions, ensuring optimal solutions, particularly for large-scale instances, and implementing methodologies such as Branch and Price. Second, there is a requirement to improve the computational time required to find near-optimal solutions, which can be addressed through a matheuristic approach. Moreover, this angular covering structure can be extended by incorporating more real-world factors, such as double coverage maximization, connectivity of located facilities, stochastic demand, and capacity. Additionally, the angular covering framework can be adapted for application in other networks, including Internet and communication networks, to enhance their coverage and connectivity efficiency.

REFERENCES

[1] S. L. Hakimi, "Optimum distribution of switching centers in a communication network and some related graph theoretic problems," *Oper. Res.*, vol. 13, no. 3, pp. 462–475, Jun. 1965.

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

[3] J. M. Gleason, "A set covering approach to bus stop location," *Omega*, vol. 3, no. 5, pp. 605–608, Oct. 1975.

[4] M. Daskin, P. C. Jones, and T. J. Lowe, "Rationalizing tool selection in a flexible manufacturing system for sheet-metal products," *Oper. Res.*, vol. 38, no. 6, pp. 1104–1115, Dec. 1990.

[5] C. Wu and A. T. Murray, "Optimizing public transit quality and system access: The multiple-route, maximal covering/shortest-path problem," *Environ. Planning B, Planning Des.*, vol. 32, no. 2, pp. 163–178, Apr. 2005.

[6] L. Ye, C. Ye, and Y.-F. Chuang, "Location set covering for waste resource recycling centers in Taiwan," *Resour., Conservation Recycling*, vol. 55, no. 11, pp. 979–985, Sep. 2011.

[7] H. Ye and H. Kim, "Locating healthcare facilities using a network-based covering location problem," *GeoJournal*, vol. 81, no. 6, pp. 875–890, Dec. 2016.

[8] N. R. Paul, B. J. Lunday, and S. G. Nurre, "A multiobjective, maximal conditional covering location problem applied to the relocation of hierarchical emergency response facilities," *Omega*, vol. 66, pp. 147–158, Jan. 2017.

[9] E. Álvarez-Miranda, M. Goycoolea, I. Ljubić, and M. Sinnl, "The generalized reserve set covering problem with connectivity and buffer requirements," *Eur. J. Oper. Res.*, vol. 289, no. 3, pp. 1013–1029, Mar. 2021.

[10] S. S. V. Vianna, "The set covering problem applied to optimisation of gas detectors in chemical process plants," *Comput. Chem. Eng.*, vol. 121, pp. 388–395, Feb. 2019.

[11] Y. Park, P. Nielsen, and I. Moon, "Unmanned aerial vehicle set covering problem considering fixed-radius coverage constraint," *Comput. Operations Res.*, vol. 119, Jul. 2020, Art. no. 104936.

[12] J. W. Chrissis, R. P. Davis, and D. M. Miller, "The dynamic set covering problem," *Appl. Math. Model.*, vol. 6, no. 1, pp. 2–6, 1982.

[13] J. R. Current and J. E. Storbeck, "Capacitated covering models," *Environ. Planning B, Planning Des.*, vol. 15, pp. 153–163, Jun. 1988.

[14] R. Love, J. Morris, and G. O. Wesolowsky, *Facility Location: Models and Methods*. New York, NY, USA: North-Holland, 1988.

[15] M. J. Hwang, C. I. Chiang, and Y. H. Liu, "Solving a fuzzy set-covering problem," *Math. Comput. Model.*, vol. 40, nos. 7–8, pp. 861–865, Oct. 2004.

[16] Y. Ni, "Fuzzy minimum weight edge covering problem," *Appl. Math. Model.*, vol. 32, no. 7, pp. 1327–1337, Jul. 2008.

[17] L. S. Pessoa, M. G. C. Resende, and C. C. Ribeiro, "A hybrid Lagrangean heuristic with GRASP and path-relinking for set k-covering," *Comput. Oper. Res.*, vol. 40, no. 12, pp. 3132–3146, Dec. 2013.

[18] M. Daskin, *Covering Problems*. Hoboken, NJ, USA: Wiley, 2013, ch. 4, pp. 124–192.

[19] O. Berman, Z. Drezner, and D. Krass, "Generalized coverage: New developments in covering location models," *Comput. Oper. Res.*, vol. 37, pp. 1675–1687, Oct. 2010.

[20] S. J. Hosseininezhad, M. S. Jabalameli, and S. G. J. Naini, "A continuous covering location model with risk consideration," *Appl. Math. Model.*, vol. 37, no. 23, pp. 9665–9676, Dec. 2013.

[21] A. Ebrahimi-Zade, H. Hosseini-Nasab, Y. Zare-Mehrjerdi, and A. Zahmatkesh, "Multi-period hub set covering problems with flexible radius: A modified genetic solution," *Appl. Math. Model.*, vol. 40, no. 4, pp. 2968–2982, Feb. 2016.

[22] B. Zhang, J. Peng, and S. Li, "Covering location problem of emergency service facilities in an uncertain environment," *Appl. Math. Model.*, vol. 51, pp. 429–447, Nov. 2017.

[23] K. Yabuta and H. Kitazawa, "Optimum camera placement considering camera specification for security monitoring," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2008, pp. 2114–2117.

[24] Y.-G. Fu, J. Zhou, and L. Deng, "Surveillance of a 2D plane area with 3D deployed cameras," *Sensors*, vol. 14, no. 2, pp. 1988–2011, Jan. 2014.

[25] Z. Han, S. Li, C. Cui, H. Song, Y. Kong, and F. Qin, "Camera planning for area surveillance: A new method for coverage inference and optimization using location-based service data," *Comput., Environ. Urban Syst.*, vol. 78, Nov. 2019, Art. no. 101396.

[26] M. S. S. Suresh, A. Narayanan, and V. Menon, "Maximizing camera coverage in multicamera surveillance networks," *IEEE Sensors J.*, vol. 20, no. 17, pp. 10170–10178, Sep. 2020.

[27] 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," *Comput., Environ. Urban Syst.*, vol. 88, Jul. 2021, Art. no. 101638.

[28] S. Saffari and Y. Fathi, "Set covering problem with conflict constraints," *Comput. Operations Res.*, vol. 143, Jul. 2022, Art. no. 105763.

[29] L. Chen, S.-J. Chen, W.-K. Chen, Y.-H. Dai, T. Quan, and J. Chen, "Efficient presolving methods for solving maximal covering and partial set covering location problems," *Eur. J. Oper. Res.*, vol. 311, no. 1, pp. 73–87, Nov. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221723003417

[30] H. Jang and T. Lee, "Demand point aggregation method for covering problems with gradual coverage," *Comput. Oper. Res.*, vol. 60, pp. 1–13, Aug. 2015.

[31] M. A. Pereira, L. A. N. Lorena, and E. L. F. Senne, "A column generation approach for the maximal covering location problem," *Int. Trans. Oper. Res.*, vol. 14, no. 4, pp. 349–364, Jul. 2007.

[32] F. D. A. Corrěa, L. A. N. Lorena, and G. M. Ribeiro, "A decomposition approach for the probabilistic maximal covering location-allocation problem," *Comput. Operations Res.*, vol. 36, no. 10, pp. 2729–2739, Oct. 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054808002475

[33] L. Alfandari, J. Sadki, A. Plateau, and A. Nagih, "Hybrid column generation for large-size covering integer programs: Application to transportation planning," *Comput. Oper. Res.*, vol. 40, no. 8, pp. 1938–1946, Aug. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054813000609

[34] T. Larsson and N.-H. Quttineh, "Column generation extensions of set covering greedy heuristics," *Oper. Res. Lett.*, vol. 50, no. 6, pp. 738–744, Nov. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167637722001407

[35] M. Muniz and T. Flamand, "A column generation approach for the team formation problem," *Comput. Oper. Res.*, vol. 161, 2024, Art. no. 106406. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054823002708

[36] G. Desaulniers, J. Desrosiers, and M. Solomon, *Column Generation*, vol. 5. New York, NY, USA: Springer, 2006.

FREDY BARRIGA-GALLEGOS was born in Bulnes, Chile, in 1995. He received the B.S. and M.S. degrees in Industrial Engineering from Universidad Catolica de la Santísima Concepcion. He is currently pursuing the Ph.D. degree in Industrial Engineering with Pontificia Universidad Católica de Valparaíso. His research interests include logistics, network design, and facility location.

ARMIN LÜER-VILLAGRA received the Ph.D. degree in engineering science from Pontificia Universidad Catolica de Chile. He is currently an Associate Professor with the Department of Engineering Sciences, Universidad Andres Bello, Talcahuano, Chile. His research interests include location analysis, hub-and-spoke, network design, and exact and approximate mathematical optimization procedures.

GABRIEL GUTIÉRREZ-JARPA was born in Concepcion, Chile, in 1975. He received the M.Sc. and Ph.D. degrees in engineering sciences from Pontificia Universidad Catolica de Chile, in 2009. From 2007 to 2009, he was an Assistant Professor at Universidad Catolica de la Santisima Concepcion, Concepcion. Since 2014, he has been an Associate Professor with the School of Industrial Engineering, Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile. His research interests include vehicle routing, logistics, and rapid network design. He received the title of Industrial Engineer from Universidad del Bio-Bio, Concepcion, in 2001.