



Quadratic p-Median Formulations with Connectivity Costs Between Facilities

Cesar Sandoval¹ , Pablo Adasme¹  , and Ali Dehghan Firoozabadi² 

¹ Department of Electrical Engineering, Universidad de Santiago de Chile,
Avenida Ecuador 3519, Santiago, Chile

{cesar.sandoval.n,pablo.adasme}@usach.cl

² Department of Electricity, Universidad Tecnológica Metropolitana,
Av. Jose Pedro Alessandri 1242, 7800002 Santiago, Chile

adehghanfiroozabadi@utem.cl

Abstract. In this paper, we consider the problem of minimizing simultaneously the total connectivity cost of a set of users to a set of facility nodes and between facilities themselves. The problem arises as an extension of the p-Median problem, which is a classical combinatorial optimization problem. We propose two mixed-integer quadratic programming models which allow obtaining optimal solutions. Example application domains where the proposed models can be utilized include modeling and management of smart transportation and wireless networks, to name a few. Our first model is derived as an extension of the classical p-Median formulation. Whereas the second one is an alternative set-covering model. Finally, we linearize these models and strengthen them by imposing additional linearized quadratic cuts. We solve hard instances with up to 60 facility nodes and 350 users with the Gurobi solver. Our preliminary numerical results indicate that the linear set-covering formulation allows solving all tested instances to optimality in significantly less computational effort, which is not possible to achieve with the other proposed models.

Keywords: Mixed-integer quadratic and linear programming · p-Median problem · Smart transportation and Wireless networks applications

1 Introduction

The p-Median problem is a classical combinatorial optimization problem which can be described as follows. First, we are given a set of facility nodes \mathcal{J} and a set of users (customers) \mathcal{I} with cardinalities n and m , respectively. Then, the problem consists of finding an optimal subset of facility nodes $\mathcal{P} \subseteq \mathcal{J}$ with cardinality p that minimizes the total connectivity cost in such a way that each user of \mathcal{I} is assigned to a unique facility node in \mathcal{P} . It is a discrete type location problem with many real-life applications including the location of industrial plants, warehouses, public facilities, and network design problems, to name a few

[1,8]. In particular, for network design problems, it provides a useful modeling framework as it allows forming backbone network structures [1,2,8,9].

In this paper, we consider the problem of minimizing simultaneously the total connectivity cost of users to facilities, as in the classical p -Median problem, plus the total connectivity cost between pairs of facilities. Notice that the latter cost is not usually incorporated in the classical p -Median problem. We argue that the motivation for doing so can be justified within many application domains. For example, when moving patients from one hospital to another one under pandemic situations like covid-19. Similarly, in a wireless network, it is highly required that the facilities forming a backbone network structure are installed as close as possible since this would allow more rapid transmission of the data and at lower power consumption. Notice that even though the proposed models provide a generic modeling framework, they can be straightforwardly applied and adapted to many optimization problems related to smart transportation and wireless networks [1]. Also notice that in a graph layout configuration of a network, the connectivity cost between two facilities can be computed as a function of the shortest path distance between them [5]. We propose two mixed-integer quadratic programming models. The first one is derived as an extension of the classical p -Median formulation. Whereas the second one is formulated based on an alternative set-covering formulation related to the p -Median problem [3]. Then, we apply the Fortet linearization method [6] and obtain two additional mixed-integer linear programming (MIP) models. Finally, each linearized model is further strengthened by imposing additional linearized quadratic cuts. In particular, we derive a cardinality cut that allows tightening the linear programming relaxations significantly.

As far as we know, our proposed models are new to the literature. A similar problem was studied in [10] where the author reports a new integer quadratic formulation of a general hub-location problem. The author discusses a variety of alternative solution strategies and proposes two heuristics for the task of siting 2, 3, or 4 hubs to serve interactions between sets of 10, 15, 20, and 25 cities. Other related works, from the domain of wireless networks and urban planning, where our proposed formulations can be applied, are reported in references [7,11]. In particular, in [7], the author proposes a distribution location plan for multi-sink nodes in a wireless sensor network based on the p -median problem. The author also proposes a heuristic algorithm and shows that the proposed strategy can effectively reduce the overall energy consumption, improve the efficiency of the network service and extend the network lifetime. Analogously, in [11], the authors incorporate big data in urban planning for better modeling of urban dynamics and more efficiently allocating limited resources. The authors propose a high-performance computing-based algorithm based on random sampling and spatial voting techniques to solve large-sized p -median problems. Numerical results show that their proposed algorithm provides high-quality solutions and reduces computing time significantly. They also demonstrate the dynamic scalability of their proposed algorithm.

The paper is organized as follows. In Sect. 2, we present and explain the quadratic and linear mathematical formulations. Then, we explain how we derive quadratic cuts, which allow tightening each of the linear programming (LP) relaxations. Subsequently, in Sect. 3, we conduct preliminary numerical experiments in order to compare all the proposed models. Finally, in Sect. 4, we present our main conclusions and provide some insight for future research.

2 Mathematical Formulations and Quadratic Cuts

In this section, we first present and explain the proposed mixed-integer quadratic and linear models. Then, we describe how we obtain the quadratic cuts, which are added to the MIP models.

2.1 Proposed Mathematical Formulations

In order to write a first quadratic model, we define the following binary variables

$$y_j = \begin{cases} 1, & \text{if facility node } j \in \mathcal{J} \text{ belongs to subset } \mathcal{P}. \\ 0, & \text{otherwise.} \end{cases}$$

and

$$x_{ij} = \begin{cases} 1, & \text{if user } i \in \mathcal{I} \text{ is assigned to facility node } j \in \mathcal{J}. \\ 0, & \text{otherwise.} \end{cases}$$

We also define the input matrices $C = (C_{ij})$ for all $i \in \mathcal{I}$ and $j \in \mathcal{J}$, and $D = (D_{ij})$ for all $i, j \in \mathcal{J}$ as the required distance matrices. Notice that each entry in each of these input matrices represents the distance between elements i and j . Consequently, a first model can be written as follows

$$Q_1 : \min_{\{x,y\}} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} C_{ij} x_{ij} + \sum_{i \in \mathcal{I}} \sum_{\substack{j \in \mathcal{J} \\ (i \neq j)}} \frac{1}{2} D_{ij} y_i y_j \quad (1)$$

$$\text{s.t.: } \sum_{j \in \mathcal{J}} x_{ij} = 1, \quad \forall i \in \mathcal{I} \quad (2)$$

$$\sum_{j \in \mathcal{J}} y_j = p \quad (3)$$

$$x_{ij} \leq y_j, \quad \forall i \in \mathcal{I}, j \in \mathcal{J} \quad (4)$$

$$x \in \{0, 1\}^{mn}, y \in \{0, 1\}^n \quad (5)$$

where the first and second terms in the objective function (1) represent the connectivity costs between users and facilities, and between facilities themselves, respectively. The constraints (2) ensure that each user $i \in \mathcal{I}$ is connected to a unique facility node $j \in \mathcal{J}$. Next, the constraint (3) is used to select p out of n facilities from \mathcal{J} . Subsequently, the constraints (4) ensure that each user $i \in \mathcal{I}$ is connected to a facility node $j \in \mathcal{J}$ if and only if j is active. Finally, constraints (5) are the domain constraints for the decision variables.

In order to obtain an equivalent MIP formulation for Q_1 , we apply the Fortet linearization method [6]. This method consists of replacing each quadratic term in the objective function (1) with a new variable $z_{ij} = y_i y_j$ for all $i, j \in \mathcal{J}$, ($i < j$) while simultaneously adding linearization constraints. This leads to the following equivalent MIP model

$$M_1 : \min_{\{x,y,z\}} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} C_{ij} x_{ij} + \sum_{i \in \mathcal{J}} \sum_{\substack{j \in \mathcal{J} \\ (i < j)}} D_{ij} z_{ij} \quad (6)$$

$$\text{s.t.: (2) - (4)}$$

$$z_{ij} \leq y_i, \quad \forall i, j \in \mathcal{J}, (i < j) \quad (7)$$

$$z_{ij} \leq y_j, \quad \forall i, j \in \mathcal{J}, (i < j) \quad (8)$$

$$z_{ij} \geq y_i + y_j - 1, \quad \forall i, j \in \mathcal{J}, (i < j) \quad (9)$$

$$x \in \{0, 1\}^{mn}, y \in \{0, 1\}^n, z \in \{0, 1\}^{\frac{n(n-1)}{2}}, \quad (10)$$

where the objective function (6) is no longer a quadratic one, but a linear one. The constraints (7)–(9) are the linearization constraints and ensure that variable $z_{ij} = 1$ if and only if both $y_i = y_j = 1$ for all $i, j \in \mathcal{J}$, ($i < j$). Otherwise, if either y_i or y_j equals zero, then $z_{ij} = 0$. Notice that we do not use both variables z_{ij} and z_{ji} in M_1 as each connection link is considered only once.

In order to write an equivalent set-covering formulation, we first add an artificial facility node to set \mathcal{J} . Denote this new set by $\hat{\mathcal{J}}$. Next, we define an additional input matrix $\hat{C} = (\hat{C}_{ij})$ for each $i \in \mathcal{I}$ and $j \in \hat{\mathcal{J}}$. This new extended matrix is constructed as follows. First, let \hat{C} be an empty matrix. Then, for each $i \in \mathcal{I}$, we sort in ascending order the corresponding row vector of matrix $C = (C_{ij})$, for all $j \in \mathcal{J}$ and add the resulting sorted vector to the i th row of matrix \hat{C} . Finally, we add an extra zero column vector to the left of matrix \hat{C} . Similarly, we define the input symmetric matrix $\hat{D} = (\hat{D}_{ij})$ for all $i, j \in \hat{\mathcal{J}}$. This matrix \hat{D} is constructed by adding zero column and row vectors to the left and at the top of matrix D , respectively. To terminate, we define the following cumulative variables

$$w_{ij} = \begin{cases} 1, & \text{if the distance cost of user } i \in \mathcal{I} \text{ is at least } \hat{C}_{ij}, \text{ for all } j \in \hat{\mathcal{J}} \\ & \text{no matter which median it is allocated to.} \\ 0, & \text{otherwise.} \end{cases}$$

Thus, a quadratic p-Median formulation can now be written as

$$Q_2 : \min_{\{w,y\}} \sum_{i \in \mathcal{I}} \sum_{j \in \hat{\mathcal{J}} \setminus \{1\}} (\hat{C}_{ij} - \hat{C}_{i,j-1}) w_{ij} + \sum_{i \in \hat{\mathcal{J}} \setminus \{1\}} \sum_{\substack{j \in \hat{\mathcal{J}} \setminus \{1\} \\ (i \neq j)}} \frac{1}{2} \hat{D}_{ij} y_i y_j \quad (11)$$

$$\text{s.t.: } \sum_{j \in \hat{\mathcal{J}} \setminus \{1\}} y_j = p \quad (12)$$

$$w_{ik} + \sum_{\substack{j \in \hat{\mathcal{J}} \setminus \{1\} \\ \{C_{ij} < \hat{C}_{ik}\}}} y_j \geq 1, \quad \forall i \in \mathcal{I}, k \in \hat{\mathcal{J}} \setminus \{1\} \quad (13)$$

$$w \in \{0, 1\}^{m(n+1)}, y \in \{0, 1\}^{n+1} \quad (14)$$

Analogously as for Q_1 , we can obtain its linearized version as

$$M_2 : \min_{\{w, y, z\}} \sum_{i \in \mathcal{I}} \sum_{j \in \hat{\mathcal{J}} \setminus \{1\}} (\hat{C}_{ij} - \hat{C}_{i,j-1}) w_{ij} + \sum_{i \in \hat{\mathcal{J}} \setminus \{1\}} \sum_{\substack{j \in \hat{\mathcal{J}} \setminus \{1\} \\ (i < j)}} \hat{D}_{ij} z_{ij} \quad (15)$$

s.t.: (12) – (13)

$$z_{ij} \leq y_i, \quad \forall i, j \in \hat{\mathcal{J}} \setminus \{1\}, (i < j) \quad (16)$$

$$z_{ij} \leq y_j, \quad \forall i, j \in \hat{\mathcal{J}} \setminus \{1\}, (i < j) \quad (17)$$

$$z_{ij} \geq y_i + y_j - 1, \quad \forall i, j \in \hat{\mathcal{J}} \setminus \{1\}, (i < j) \quad (18)$$

$$w \in \{0, 1\}^{m(n+1)}, y \in \{0, 1\}^{n+1}, z \in \{0, 1\}^{\frac{(n+1)n}{2}} \quad (19)$$

Notice that the Fortet linearization constraints (16)–(18) are now imposed in M_2 for all $i, j \in \hat{\mathcal{J}} \setminus \{1\}$.

2.2 Quadratic Cuts Used to Strengthen the MIP Models

In this subsection, we explain how we obtain quadratic cuts, which are then linearized to further strengthen the LP relaxations of the MIP models. By doing so, we intend to measure the impact of these cuts on the performance of the branch and cut algorithm of the Gurobi solver [4] when solving M_1 and M_2 . We denote by M_1^s and M_2^s the strengthened versions of models M_1 and M_2 , respectively. The set of linearized quadratic cuts we add in M_1^s are

$$\sum_{j \in \mathcal{J}} z_{ij} = p y_i, \quad \forall i \in \mathcal{J} \quad (20)$$

$$\sum_{\substack{i, j \in \mathcal{J} \\ (i < j)}} z_{ij} = \frac{p(p-1)}{2} \quad (21)$$

$$y_i + y_j \leq 1 + z_{ij}, \quad \forall i, j \in \mathcal{J}, (i < j) \quad (22)$$

The first cut (20) is obtained by multiplying constraint (3) by each y_i , for all $i \in \mathcal{J}$. Whilst the cardinality cut (21) is obtained by multiplying constraint (3) with each variable y_i for all $i \in \mathcal{J}$. Then, we take the sum over all the resulting constraints and obtain the equality $\sum_{i, j \in \mathcal{J}} y_i y_j = p^2$, which is equivalent to

$$\sum_{i, j \in \mathcal{J}, (i \neq j)} y_i y_j = p(p-1)$$

and

$$\sum_{i,j \in \mathcal{J}, (i < j)} z_{ij} = \frac{p(p-1)}{2}$$

Subsequently, the cuts in (22) are obtained from the quadratic valid inequalities $(1 - y_i)(1 - y_j) \geq 0$ for all $i, j \in \mathcal{J}, (i < j)$. Observe that each term in these inequalities is nonnegative since $y_i \leq 1$ for each $i \in \mathcal{J}$. Finally, notice that the cuts (20)–(22) can be straightforwardly adapted for M_2 while using the set $\hat{\mathcal{J}} \setminus \{1\}$ instead of \mathcal{J} .

3 Numerical Experiments

In this section, we present preliminary numerical results for all the proposed models. For this purpose, we implement a Python program using Gurobi solver [4]. In particular, we solve each quadratic model while setting the Gurobi option parameter *Nonconvex* to a value of 2. This option allows one to solve a non-convex quadratic problem with spatial branch and bound algorithms. We decided to use this option as it proved to be the most effective one in terms of CPU time required to solve the quadratic models. Whilst for the MIP and LP models, we use the Gurobi solver with default options. This implies solving the linear models with state-of-the-art branch and cut-based algorithms. The numerical experiments have been carried out on an Intel(R) 64 bits core (TM) with 3 GHz and 8G of RAM under Windows 10. So far, we generate eight instances with dimensions of $n = \{50; 60\}$ and $m = \{200; 250; 300; 350\}$, and each of them is solved for values of $P = \{10; 20\}$. We mention that this range of parameter values was arbitrarily chosen in order to ensure that hard instances were obtained. In a larger version of this paper, a wider range of parameter values will be considered. The input matrices $C = C_{uj}$ and $D = D_{ij}$ for all $u \in \mathcal{I}$ and $i, j \in \mathcal{J}$ are generated by computing the distances between users and facilities and between facilities themselves. Notice that matrix D is symmetric. The coordinates of each user and facility node are randomly drawn from the interval $(0; 1)$.

In Table 1, we present preliminary numerical results obtained with models Q_1 and M_1 . More precisely, in columns 1–4 we present the instance number, and the values of p , m , and n , respectively. Next, in columns 5–8 we report for Q_1 , the best objective function value obtained in at most 1h of CPU time, the number of branch and bound nodes, CPU time in seconds required to solve the model, and the *MipGap* parameter value of Gurobi solver, respectively. Gurobi computes this parameter by subtracting from the best incumbent solution the best lower bound obtained. Then, it divides this result by the incumbent solution again. Notice that this value is reported as a percentage. Also notice that if the Gurobi solver terminates with an optimal solution, then the *MipGap* parameter should be equal to zero. Subsequently, in columns 9–15 we report for M_1 , the best objective function value obtained in at most 1 h, the number of branch and bound nodes, CPU time in seconds, the optimal objective value of its LP relaxation, and CPU time in seconds required to solve it, and the gap and *MipGap* values, respectively. The gap is computed by $\left[\frac{Best-LP}{Best} \right] * 100$.

Table 1. Numerical results obtained with Q_1 and M_1 .

#	P	m	n	Q_1				M_1						
				Best	$B\&Bn$	CPU(s)	MipGap%	Best	$B\&Bn$	CPU(s)	LP	CPU(s)	Gap%	MipGap%
1	10	200	50	45.76	59834	2133.63	0	45.76	10494	1015.92	24.17	0.43	47.18	0
2		250		53.59	45816	2090.6	0	53.59	6398	1156.28	31.15	0.58	41.86	0
3		300		60.54	36938	1964.24	0	60.54	4850	1246.64	37.34	0.61	38.31	0
4		350		66.97	26966	1824.55	0	66.97	2456	703.56	43.96	0.83	34.35	0
5	10	200	60	45.55	50480	3600	0.06	45.55	19046	3229.7	24.03	0.55	47.24	0
6		250		53.35	46754	3600	0.05	53.35	11860	3600	31.04	0.72	41.8	0.03
7		300		60.39	41313	3600	0.05	60.39	11310	3600	37.26	0.81	38.3	0.03
8		350		66.77	33585	3600	0.05	66.77	10374	3600	43.92	0.96	34.21	0.02
1	20	200	50	88.19	2595	962.54	0	88.19	233	120.82	18.4	0.19	79.13	0
2		250		96.21	5095	2129.14	0	96.21	521	375.69	23.47	0.36	75.6	0
3		300		104.17	7559	3600	0.02	104.17	1725	1118.44	28.44	0.5	72.69	0
4		350		110.67	5372	3600	0.04	110.67	1457	1062.42	33.29	0.76	69.91	0
5	20	200	60	84.91	4278	3600	0.06	84.91	1153	858.49	17.81	0.29	79.02	0
6		250		93.24	3566	3600	0.09	93.24	1140	1473.24	22.83	0.55	75.51	0
7		300		101.44	2966	3600	0.12	101.44	1574	3600	27.7	0.67	72.68	0.02
8		350		108.47	2771	3600	0.13	108.36	1513	3600	32.49	0.9	70.01	0.03

Table 2. Numerical results obtained with Q_2 and M_2 .

#	P	m	n	Q_2				M_2						
				Best	$B\&Bn$	CPU(s)	MipGap%	Best	$B\&Bn$	CPU(s)	LP	CPU(s)	Gap%	MipGap%
1	10	200	50	45.76	5799	1122.08	0	45.76	188	91.79	24.17	0.77	47.18	0
2		250		53.59	5963	1499.08	0	53.59	124	113.52	31.15	1.29	41.86	0
3		300		60.54	2050	767.52	0	60.54	1	45.86	37.34	1.52	38.31	0
4		350		66.97	2101	872.13	0	66.97	1	37.74	43.96	1.89	34.35	0
5	10	200	60	45.55	9298	3600	0.04	45.55	1196	934.09	24.03	1.24	47.24	0
6		250		53.35	7716	3600	0.05	53.35	791	767.94	31.04	1.79	41.8	0
7		300		60.67	5756	3600	0.04	60.39	312	424.37	37.26	2.44	38.3	0
8		350		66.77	4642	3600	0.02	66.77	1	145.03	43.92	2.81	34.21	0
1	20	200	50	88.19	1383	727.43	0	88.19	17	44.02	18.4	0.43	79.13	0
2		250		96.21	1795	998.88	0	96.21	145	73.65	23.47	0.58	75.6	0
3		300		104.17	3036	1721.66	0	104.17	61	84.32	28.44	0.81	72.69	0
4		350		110.67	4148	2266.1	0	110.67	116	110.93	33.29	1.15	69.91	0
5	20	200	60	84.91	2962	3600	0.04	84.91	43	164.47	17.81	0.62	79.02	0
6		250		93.24	2456	3600	0.06	93.24	325	404.15	22.83	1.07	75.51	0
7		300		101.53	2701	3600	0.07	101.44	635	1118.62	27.7	1.24	72.68	0
8		350		108.42	2229	3600	0.08	108.35	816	1909.16	32.49	1.71	70.01	0

In Table 2, we report numerical results for the models Q_2 and M_2 . The column information of Table 2 is analogous to Table 1. Finally, in Table 3 we report preliminary numerical results for the linear models M_1^s and M_2^s . Its legend is analogous as for the linear models reported in Tables 1 and 2 for M_1 and M_2 , respectively.

From Table 1, we first observe that both models Q_1 and M_1 allow obtaining the same objective function values except for the instance #8, for which we obtain a slightly smaller value with M_1 . Regarding the number of branch and bound nodes, we see that Gurobi reports a significantly less number of nodes

Table 3. Numerical results obtained with M_1^s and M_2^s .

#	P	m	n	M_1^s						M_2^s							
				Best	B&Bn	CPU(s)	LP	CPU(s)	Gap%	MipGap%	Best	B&Bn	CPU(s)	LP	CPU(s)	Gap%	MipGap%
1	10	200	50	45.76	7702	928.61	34.63	0.41	24.32	0	45.76	158	115.51	34.63	1.39	24.32	0
2		250		53.59	5063	1105.62	41.62	0.48	22.33	0	53.59	9	75.24	41.62	1.46	22.33	0
3		300		60.54	3644	1008.27	48.26	0.71	20.27	0	60.54	1	46.72	48.26	1.76	20.27	0
4		350		66.97	4062	1107.03	54.79	0.88	18.18	0	66.97	1	42.38	54.79	1.98	18.18	0
5	10	200	60	45.55	16213	3600	33.48	0.7	26.48	0.01	45.55	1062	869.88	33.48	2.03	26.48	0
6		250		53.35	10270	3600	40.5	0.65	24.08	0.03	53.35	1068	1454.03	40.5	2.09	24.08	0
7		300		60.39	8741	3600	47.2	0.83	21.85	0.04	60.39	257	520.99	47.2	2.6	21.85	0
8		350		66.77	7240	3600	53.77	0.99	19.45	0.02	66.77	1	146.25	53.77	2.98	19.45	0
1	20	200	50	88.19	876	221.09	71.69	0.35	18.7	0	88.19	1	41.15	71.69	1.34	18.7	0
2		250		96.21	1088	368.09	76.93	0.39	20.03	0	96.21	1	52.63	76.93	1.55	20.03	0
3		300		104.17	1327	768.92	82.22	0.46	21.07	0	104.17	1	77.44	82.22	2.0	21.07	0
4		350		110.67	3674	2335.46	87.34	0.55	21.07	0	110.67	55	88.82	87.34	2.33	21.07	0
5	20	200	60	84.91	1156	683.22	66.37	0.46	21.83	0	84.91	1	118.39	66.37	1.76	21.83	0
6		250		93.24	2254	1905.38	71.68	0.59	23.12	0	93.24	1	175.0	71.68	2.4	23.12	0
7		300		101.44	2545	3600	76.92	0.73	24.17	0.02	101.44	223	482.71	76.92	2.9	24.17	0
8		350		108.35	1903	3600	82.08	0.75	24.24	0.03	108.35	488	866.76	82.08	3.31	24.24	0

for M_1 . The CPU times show that we can obtain the optimal solution of the problem for a larger number of instances with M_1 in less than 1h. Next, we observe that the *MipGap* values reported for both models are close to zero. The latter indicates that the solutions reported are near-optimal. Notice that from these values, we can obtain tight lower bounds for the problem as well. Finally, we see that the LP relaxation of M_1 is not tight when compared to the best objective function values and that the gaps obtained increase with p .

From Table 2, we observe similar trends for Q_2 and M_2 . We notice that M_2 significantly outperforms model Q_2 in terms of the best objective function and CPU time values obtained. Indeed, we obtain the optimal solution to the problem with proven optimality for all tested instances. This cannot be achieved by any of the other proposed models reported in Tables 1 and 2. This achievement can also be observed by looking at the number of branch and bound nodes obtained, which are significantly lower for M_2 . Finally, we observe that the LP relaxation of M_2 is not tight either. Similarly, from Table 3 we observe that M_2 outperforms M_1 in terms of all column information reported. In particular, we confirm that M_2 allows obtaining the optimal solution of the problem for all the instances. Perhaps, one of the most interesting observations of Table 3 is that the LP bounds obtained with M_2^s are significantly higher than those reported in Tables 1 and 2, respectively. Ultimately, we observe that the impact of adding linearized quadratic cuts in M_2 is not so strong in terms of CPU time values obtained. However, we see that for most of the instances, the number of branch and bound nodes decreases when these cuts are added to the models.

4 Conclusions

In this paper, we considered the problem of minimizing simultaneously the total connectivity cost of a set of users to a set of facility nodes and among facilities themselves. The problem is motivated by the potential development of future smart transportation and wireless network applications where

interacting facilities will be required to be connected and nearly reachable. We proposed two mixed-integer quadratic programming models, which are further linearized. The first one is obtained as an extension of the classical p-Median formulation. Whereas the second one corresponds to an alternative set-covering model. Finally, we strengthen the linear models by imposing additional linearized quadratic cuts. So far, we solved hard instances with up to 60 facility nodes and 350 users. Our preliminary numerical results indicated that the linearized set-covering formulation outperforms all the other proposed models as it allows solving all the instances with proven optimality and in significantly less computational effort.

As future research, new models and algorithms should be investigated in order to tackle this hard combinatorial optimization problem. Complementarily, new cutting plane approaches should be proposed in conjunction with the new models. This would allow a complete validation of the proposal. Ultimately, the study problem in this paper should be adapted to more specific transportation and telecommunication network problems as part of future work.

Acknowledgments. The authors acknowledge the financial support from Projects: FONDECYT No. 11180107 and FONDECYT No. 3190147.

References

1. Adasme, P.: p-Median based formulations with backbone facility locations. *Appl. Soft Comput.* **67**, 261–275 (2018)
2. Adasme, P., Andrade, R., Lisser, A.: Minimum cost dominating tree sensor networks under probabilistic constraints. *Comput. Netw.* **112**, 208–222 (2017)
3. García, S., Labbé, M., Marín, A.: Solving large p-median problems with a radius formulation. *INFORMS J. Comput.* **23**(4), 546–556 (2011)
4. Achterberg, T.: Gurobi solver. <https://www.gurobi.com/>
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*. MIT Press and McGraw-Hill, Cumberland (2009)
6. Fortet, R.: Applications de l'algèbre de boole en recherche operationelle. *Rev. Fr. Rech. Oper.* **4**, 17–26 (1960)
7. Haicheng, L.: Research on distribution of sink nodes in wireless sensor network. In: 2010 Third International Symposium on Information Science and Engineering, pp. 122–124 (2010). <https://doi.org/10.1109/ISISE.2010.56>
8. Hakimi, S.L.: Optimum location of switching centers and the absolute centers and medians of a graph. *Oper. Res.* **12**, 450–459 (1964)
9. Ho-Yin, M., Zuo-Jun, M.S.: Integrated Modeling for Location Analysis. In: *Now Foundations and Trends*, p. 164 (2016)
10. O'Kelly, M.E.: A quadratic integer program for the location of interacting hub facilities. *Eur. J. Oper. Res.* **32**, 393–404 (1987)
11. Wangshu, M., Daoqin, T.: On solving large p-median problems. *Environ. Planning B Urban Anal. City Sci.* **47**(6), 981–996 (2020)