A BRANCH-AND-CUT ALGORITHM FOR A BUDGET CONSTRAINED CENTROID PROBLEM

Marcos Costa Roboredo
Production Engineering Department - Fluminense Federal University
Rua Passo da Pátria 156, 24210-240, Niterói, RJ, Brazil
mcr.marcos@yahoo.com.br

Artur Alves Pessoa
Production Engineering Department - Fluminense Federal University
Rua Passo da Pátria 156, 24210-240, Niterói, RJ, Brazil
artur@producao.uff.br

RESUMO

Nós tratamos um problema centróide discreto composto de duas firmas, líder e seguidora, competindo para atender a demanda de clientes em um determinado mercado. A demanda de cada cliente pode ser totalmente atendida por uma das firmas ou não atendida de acordo com uma regra de distância. A líder localiza suas facilidades sobre um grafo sabendo que a seguidora reagirá fazendo o mesmo e deseja maximizar sua fatia do mercado no pior caso. O número de facilidades localizadas pelas duas firmas respeita uma restrição orçamentária. O problema consiste em decidir onde a líder localiza suas facilidades. Este problema é $\sum_2^p$-difícil. Apesar disso, nós apresentamos uma formulação por programação inteira com um número polinomial de variáveis e um número exponencial de restrições. Nós reportamos diversos experimentos variando o número de clientes e facilidades e os valores de orçamento. Nossos resultados mostram que nosso método resolve em torno de uma hora instâncias com até 225 clientes, 150 facilidades com as firmas podendo localizar até 7 facilidades.

PALAVRAS CHAVE. Líder-Seguidor, Localização de Facilidades Competitiva, Teoria dos Jogos.

Área Principal: Otimização

ABSTRACT

We deal with a discrete centroid problem composed of two noncooperative firms, a leader and a follower, competing to serve the demand of customers from a given market. Each customer's demand can be totally served by one of the two facilities or not served according to a distance rule. The leader places it facilities on a graph knowing that the follower will react by doing the same and wants to maximize it market share at the worst case. The number of facilities placed by the two firms is under a budget constraint. The problem consists of deciding where the leader places its facilities. This problem is a $\sum_2^p$-hard one. In spite of it, we present an integer programming formulation with polynomially many variables and exponentially many constraints. Moreover, we report several experiments with different number of customers and applicant facilities and different values of budgets. Our results show that our method requires less than 1 hour to solve instances with up to 225 customers and 150 applicant facilities where the firms could place up to 7 facilities.

KEY WORDS. Leader-Follower problems, Competitive Facility Location, Game Theory.

Main area: Optimization

1 Introduction

In competitive location models, two or more noncooperative firms compete to provide customers from a given market. Each customer is partially, totally or not served by facilities placed by the firms according to a customer choice rule. Different objective functions can be pursued by each firm when it decides where to place its facilities. For example, they might aim to maximize their own market share, their own profit, their own number of customers served, among others. When the decision is made sequentially, the model is called sequential competitive location model. A review of competitive location models and of sequential ones can be found in Eiselt and Laporte (1989), Friesz *et al*. (1988), Eiselt *et al*. (1993), Eiselt and Laporte (1997) and Kress and Pesch (2011).

There is a special class of sequential competitive location problems composed of two firms, leader and follower. The leader must to decide where to place its facilities knowing that the follower will react by doing the same. The problems of deciding where the leader and the follower place their facilities are respectively called centroid and medianoid problems.

We deal in this paper with a discrete centroid problem where the number of facilities placed by both the leader and the follower is under a budget constraint. Each customer's demand is fully served by the closest facility placed by either the leader or the follower if this facility is lying with a threshold distance. Otherwise, this customer is not served. The leader aims to maximize it market share at the worst case. The worst case happens when the follower places its facilities aiming to steal the maximum possible demand from the leader. We call this problem the worst case budget constrained centroid problem, or simply, WCBC centroid problem.

We found in the literature just one paper that deals specifically with the WCBC centroid problem proposed by Plastria and Vanhaverbeke (2008). The authors proposed an integer programming model for the WCBC centroid problem for the particular case where the follower places a single facility. Besides the authors also proposed models for more two cases. In the first one, the leader aims to minimize it regret while in the second one, both the leader and the follower aim to maximize their market share.

In this paper, we propose a branch-and-cut algorithm for the WCBC centroid problem. Our algorithm is based on the best exact one for the $(r|p)$-centroid problem (Hakimi, 1983) proposed by Roboredo and Pessoa (2011). The previous problem is $\sum_2^{\mathrm{p}}$-hard (Noltemeier *et al.*, 2007) and is similar to the one dealt in this paper with three differences. The first one is that both the leader and the follower have a predetermined number of facilities to place, the second one is that there is not the threshold distance while the third one is that both the leader and the follower aim to maximize their market share.

As any instance of the $(r|p)$-centroid problem is easily converted to a instance of the WCBC centroid problem, this problem is also $\sum_2^{\mathrm{p}}$-hard. Hence, that problem turns out to be harder than any optimization problem whose decision version is in NP. The hardness of that problem comes from the fact that evaluating a single leader's strategy requires to solve an NP-hard problem to decide the follower's strategy which minimizes the market share of the leader. Fortunately, this problem often spends less than one second for instances with 100 customers and 100 applicant facilities by solving an Integer Programming (IP) model.

The integer programming (IP) formulation proposed in this paper has polynomially many variables and exponentially many constraints. One should note that, since the problem is $\sum_2^{\mathrm{p}}$-hard, neither a polynomial formulation nor a formulation where all constraints can be separated in polynomial time is possible unless $NP = \sum_2^{\mathrm{p}}$. Hence, the constraints are separated during the optimization either using a greedy heuristic or exactly solving an IP model. We test our method on a group of instances randomly generated in the same way as in Plastria and Vanhaverbeke (2008). Besides we increase the difficult of the instances by increasing the budgets of the leader and the follower.

This paper is divided as follows. In Section 2, we define the problem and present an IP model for it. In section 3, we define the separation problem for the only exponential family of constraints used in our formulation, and present an IP model and a greedy heuristic for it. In section 4, we report our experiments. Finally, in section 5, we summarize our conclusions.

2 The problem

The problem is formally defined as follows: Consider two noncooperative firms (leader and follower). To place a facility i, it is necessary to spent a fixed cost f_i. The budgets of the leader and the follower are denoted respectively by B_l and B_f. The leader has to place facilities on an arena knowing that the follower will do the same. The arena is a graph $G = (V, E)$ where each vertex $v \in V$ is a customer and an applicant facility of either the leader or the follower. As a result, V is composed of the three subsets, J, L and F, where J is the set of customers and L and F are the set of applicant facilities for the leader and the follower respectively. The edge set E of G has an edge $e = (i, j)$ for each $i \in L$ and $j \in J$ and for each $i \in F$ and $j \in J$, with an associated distance d_{ij}. Each customer's demand w_j is totally served by the closest facility placed by either the leader or the follower if this facility is lying with a threshold distance δ_j. Ties are broken in favor of the follower's facilities, and ties between facilities of the same firm are broken arbitrarily. The leader aims to obtain the maximum market share at the worst case. The worst case happens when the follower places its facilities aiming to steal the maximum possible demand from the leader. The discrete worst case budget constrained centroid problem consists of deciding where the leader places its facilities.

We show that, besides its complexity, the discrete WCBC centroid problem admits an IP formulation with polynomially many variables and exponentially many constraints. Let the binary variables x_i indicate whether the leader places a facility at the location i and the continuous variables y_{ij} indicate whether the leader places the facility i trying to dominate the customer j. Finally, let the integer variable z give the total leader's market share at the worst case. Let S be the set of strategies for the follower. It means that each $S_0 \in S$ is a set of facilities under the budget constraint placed by the follower. The complete model is below.

$$\max \quad z \tag{1}$$

subject to

$$\sum_{i \in L} f_i x_i \leq B_l \tag{2}$$

$$y_{ij} \leq x_i, \qquad \forall j \in J, \forall i \in L \tag{3}$$

$$\sum_{i \in L} y_{ij} = 1, \qquad \forall j \in J \tag{4}$$

$$z \leq \sum_{j \in J} \sum_{i \in L | d_{ij} < \min\{d_{kj} | k \in S_0\} \wedge d_{ij} \leq \delta_j} w_j y_{ij}, \qquad \forall S_0 \in S \tag{5}$$

$$y_{ij} \geq 0, \qquad \forall i \in L, \forall j \in J \tag{6}$$

$$x_i \in \{0, 1\}, \qquad \forall i \in L. \tag{7}$$

The objective function (1) maximizes the total market share of the leader at the worst case. The first constraint (2) is a budget one for the facilities placed by the leader. Constraints (3) ensure the consistency between the variables x_i and y_{ij}. The set of constraints (4) indicates that for each customer j, exactly one facility i is the leader's facility nearest to j. Finally, (5) ensures that, for each possible strategy $S_0 \in S$ for the follower, the total leader's market share at the worst case z is given by the total demand associated to customers that are nearer to a leader's facility than to any location $k \in S_0$ if this leader's facility is lying with the threshold distance δ_j. Note that depending of the follower's budget B_f, (5) is composed of an exponential number of constraints. Hence, it is necessary to solve the separation problem associated to (5) in order to include in the formulation only the necessary constraints.

3 Separation Problem

In this section, we define the separation problem and two separation procedures for the constraints given by (5). Although we do not know whether this problem is NP-hard, we remark (1)-(7) result in a valid formulation for the discrete WCBC centroid problem. Hence, a polynomial algorithm for the exact separation of (5) would imply that the decision version of the discrete WCBC centroid problem belongs to NP, and thus, that $NP = \sum_2^{\text{p}}$.

The separation problem of (5) is defined as follows. Given a relaxed solution $(\overline{z}, \overline{x}, \overline{y}) \in \mathbb{R} \times [0,1]^{|L|} \times [0,1]^{|L| \times |J|}$ that satisfies (2), (3), (4) and some of the constraints (5), the separation problem consists of finding the strategy $S_0 \in S$ that maximizes the violation of (5). For that, S_0 should to minimize

$$\sum_{j \in J} w_j \sum_{i \in L | d_{ij} < \min\{d_{kj} | k \in S_0\} \wedge d_{ij} \leq \delta_j} \overline{y}_{ij}. \tag{8}$$

For each customer j, let $k_j^* = \arg\min_{k \in S_0}\{d_{kj}\}$. Then, (8) can be rewritten as

$$\sum_{j \in J} w_j \sum_{i \in L | d_{ij} \leq d_{k_j^* j} \wedge d_{ij} \leq \delta_j} \overline{y}_{ij}, \tag{9}$$

Next, we show an IP formulation for this problem. Let the binary variable s_k indicate that $k \in S_0$. Let also the binary variables t_{jk} indicate that k is the facility of S_0 that is closest to the customer j. The complete formulation is the following:

$$\min \sum_{j \in J} \sum_{k \in F} \left(w_j \sum_{i \in L | d_{ij} \leq d_{kj} \wedge d_{ij} \leq \delta_j} \overline{y}_{ij} \right) \times t_{jk} \tag{10}$$

subject to

$$\sum_{k \in F} f_k s_k \leq B_f \tag{11}$$

$$t_{jk} \leq s_k, \qquad \forall j \in J, \forall k \in F \tag{12}$$

$$\sum_{k \in F} t_{jk} = 1, \qquad \forall j \in J \tag{13}$$

$$t_{jk} \in \{0,1\}, \qquad \forall k \in F, \forall j \in J \tag{14}$$

$$s_k \in \{0,1\}, \qquad \forall k \in F. \tag{15}$$

The value of the objetive function (10) is equivalent to the sum (9). Constraint (11) ensures that S_0 respects the follower's budget. Constraints (12) ensure the consistency between the variables t_{jk} and s_k. Constraints (13) ensure that, for each customer j, there is only one facility $k \in S_0$ closest to this one.

We also propose a greedy heuristic for the separation problem. This heuristic is used to efficiently find some violated cuts avoiding some IP optimizations. To describe this heuristic, we recall that the separation problem is to find the strategy $S_0 \in S$ that minimize (8). The heuristic first greedily constructs the strategy S_0 by choosing facilities one at a time as follows. At each iteration, it chooses the facility that causes the maximum ratio between the diminution in the value of the terms of (8) and its fixed cost.

Our exact algorithm is built on the top of the branch-and-cut algorithm implemented by the CPLEX solver using the model given by (1), (2), (3), (4), (6), and (7). The constraints (5) are added through the cut callback provided by the solver by applying the separation procedures described before. All the branch-and-bound tree management is performed by CPLEX.

In order to speed up our method, the cuts are separated in the following way. We define the value of a parameter ϵ and while the gap is greater than ϵ, we separate cuts associated to the constraints (5) for any solution. Otherwise, when the gap is smaller or equal ϵ, the cuts are separated only for integer solutions (for that, we use only the IP based separation). We define the a value of ϵ according to the values of B_l and B_f where the higher the values of B_l and B_f, the lower the value of ϵ.

Based on the leader's optimal solution given by solving (1)-(7), we also obtain the maximum demand that the follower steals from the leader. That value can be obtained by solving the separation problem (10)-(15) since that problem aims to find the follower's strategy that minimizes the demand served by the leader.

4 Computational experiments

In this section, we present computational results of our method. We use the CPLEX 12.1 and all the tests are carried out in a 2.31 GHz PC AMD Phenom X4 9600 with 3 Gb of RAM. We tested our method on randomly generated instances as in Plastria and Vanhaverbeke (2008). For those instances, the coordinates of customers and facilities are randomly distributed on square grids, where each grid cell with integer coordinates is both a customer and a applicant facility for either the leader or the follower. Hence there are not common applicant facilities for the leader and for the follower. The applicant facilities for the follower (set F) is composed of those cells whose the sum of the coordinates is a multiple of 3 while the applicant facilities for the leader (set L) consists of all the other cells, so approximately $|L| \approx 2|F|$. We assumed to be euclidean the distances between customers and facilities. The customer demands w_j and the fixed costs f_i were randomly generated, uniformly distributed integer in the ranges $[50, 250]$ and $[5, 10]$ respectively. We set the following values for the budgets B_l and B_f: 15 and 25. For all the instances tested, we considered $B_l = B_f$. The maximum travel distances δ_j tested was 1, 2, $\sqrt{8}$, 3, $\sqrt{13}$, 4 and $+\infty$ where we considered that the maximum travel distances are the same for each customer. We considered 6 types of square grids: 5×5, 7×7,10×10, 12×12, 15×15 and 17×17. We set $\epsilon = 0.15$ for instances with $B_l = B_f = 15$ and $B_l = B_f = 25$ and set $\epsilon = 0.12$ for instances with $B_l = B_f = 35$

Tables 1, 2 and 3 show results of our method for $B_l = B_f = 15$, $B_l = B_f = 25$ and $B_l = B_f = 35$ respectively. The following headers are used for the columns. Columns *Grid size*,δ_j, $|J|$, $|L|$,$|F|$ indicate the instance characteristics, *Sum of demands* indicates the sum $\sum_{j \epsilon J} w_j$, *#Leader demand*,*#Follower demand* and *#Lost demand* indicate respectively the total demand served by the leader, the follower and the total demand not served at the worst case, *#Stolen demand* indicates the total demand served by the follower but it was served by the leader before the follower place its facilities, *#Cuts*, *#Nodes* and *Time(s) total* indicates respectively the total number of cuts generated, the total number of nodes created by the branch-and-cut tree and the total CPU time in seconds consumed by the complete branch-and-cut algorithm.

Note in tables 1, 2 and 3 that the total leader's market share at the worst case not decreases as the maximum travel distance or the budgets increase. Another interesting observation is that the total number of cuts and nodes is relatively small even for large instances illustrating the robustness of our method.

Table 1: Statistics of our method for $B_l = B_f = 15$

Grid size	δ_j	$\|J\|$	$\|L\|$	$\|F\|$	Sum of demands	#Leader demand	#Follower demand	#Lost demand	#Stolen demand	#Cuts Total	#B&B Nodes	#Time(s) Total
5×5	1	25	16	9	3542	853	1187	1502	484	6	10	0.3
5×5	2	25	16	9	3542	1291	1766	485	1272	4	0	0.4
5×5	$\sqrt{8}$	25	16	9	3542	1434	2108	0	1817	6	1	0.5
5×5	3	25	16	9	3542	1434	2108	0	2108	5	0	0.4
5×5	$\sqrt{13}$	25	16	9	3542	1434	2108	0	2108	5	0	0.4
5×5	4	25	16	9	3542	1434	2108	0	2108	5	0	0.5
5×5	$+\infty$	25	16	9	3542	1434	2108	0	2108	5	0	0.4
7×7	1	49	33	16	6531	1200	1389	3942	454	6	0	1.0
7×7	2	49	33	16	6531	2080	2348	2103	1113	10	20	3.3
7×7	$\sqrt{8}$	49	33	16	6531	2511	2631	1389	2046	9	25	2.7
7×7	3	49	33	16	6531	2863	2773	895	2335	8	103	4.6
7×7	$\sqrt{13}$	49	33	16	6531	3095	2980	456	2912	7	2	1.6
7×7	4	49	33	16	6531	3180	3351	0	2986	7	3	2.2
7×7	$+\infty$	49	33	16	6531	3180	3351	0	3351	12	17	7.2
10×10	1	100	67	33	14868	1736	958	12174	525	26	22	29.6
10×10	2	100	67	33	14868	4293	2975	7600	1467	11	16	7.1
10×10	$\sqrt{8}$	100	67	33	14868	5899	4473	4496	2873	8	8	10.1
10×10	3	100	67	33	14868	6709	5330	2829	3929	9	1	22.5
10×10	$\sqrt{13}$	100	67	33	14868	7182	6116	1570	4979	13	8	36.3
10×10	4	100	67	33	14868	7501	6381	986	5749	14	13	48.8
10×10	$+\infty$	100	67	33	14868	8268	6600	0	6600	15	6	78.8
12×12	1	144	96	48	21817	2043	1488	18286	696	16	13	19.4
12×12	2	144	96	48	21817	4273	3030	14514	1722	13	26	40.0
12×12	$\sqrt{8}$	144	96	48	21817	6140	4262	11415	2983	18	47	58.5
12×12	3	144	96	48	21817	7535	6031	8251	3868	15	47	85.9
12×12	$\sqrt{13}$	144	96	48	21817	8632	6565	6620	4963	11	35	86.7
12×12	4	144	96	48	21817	9140	12066	611	7383	16	25	171.9
12×12	$+\infty$	144	96	48	21817	10333	11484	0	11484	16	15	406.0
15×15	1	225	150	75	34405	1964	1563	30878	850	24	69	353.0
15×15	2	225	150	75	34405	4276	4322	25807	2142	25	113	853.3
15×15	$\sqrt{8}$	225	150	75	34405	6145	7763	20497	3936	41	111	2123.8
15×15	3	225	150	75	34405	7787	9171	17447	5346	34	127	1720.8
15×15	$\sqrt{13}$	225	150	75	34405	9263	10239	14903	6189	37	87	1977.0
15×15	4	225	150	75	34405	11625	13321	9459	8606	29	26	1332.6
15×15	$+\infty$	225	150	75	34405	15530	18875	0	18875	13	75	1125.8
17×17	1	289	192	97	43999	1996	1371	40632	472	71	449	3851.6
17×17	2	289	192	97	43999	4687	3096	36216	1783	56	418	4003.9
17×17	$\sqrt{8}$	289	192	97	43999	6728	6567	30704	3194	70	634	11425.5
17×17	3	289	192	97	43999	9168	6888	27943	3968	77	355	14644.0
17×17	$\sqrt{13}$	289	192	97	43999	10898	7693	25408	5192	63	327	9836.5
17×17	4	289	192	97	43999	13506	12599	17894	8653	47	148	7017.5
17×17	$+\infty$	289	192	97	43999	20724	23275	0	23275	27	118	5596.6

Table 2: Statistics of our method for $B_l = B_f = 25$

Grid size	δ_j	$\lvert J\rvert$	$\lvert L\rvert$	$\lvert F\rvert$	Sum of demands	#Leader demand	#Follower demand	#Lost demand	#Stolen demand	#Cuts Total	#B&B Nodes	#Time(s) Total
5×5	1	25	16	9	3429	1178	1716	535	1054	7	33	0.6
5×5	2	25	16	9	3429	1481	1888	60	1817	16	8	1.5
5×5	$\sqrt{8}$	25	16	9	3429	1481	1948	0	1948	8	14	0.8
5×5	3	25	16	9	3429	1481	1948	0	1948	8	20	1.0
5×5	$\sqrt{13}$	25	16	9	3429	1481	1948	0	1948	10	19	0.9
5×5	4	25	16	9	3429	1481	1948	0	1948	11	19	1.2
5×5	$+\infty$	25	16	9	3429	1481	1948	0	1948	11	19	1.3
7×7	1	49	33	16	6970	1995	2300	2675	828	13	44	1.9
7×7	2	49	33	16	6970	2650	3747	573	2312	21	105	11.0
7×7	$\sqrt{8}$	49	33	16	6970	2753	3436	781	3436	19	145	11.0
7×7	3	49	33	16	6970	2856	4114	0	3732	19	107	9.3
7×7	$\sqrt{13}$	49	33	16	6970	2882	4088	0	3959	21	264	17.3
7×7	4	49	33	16	6970	2882	4088	0	4088	21	192	14.2
7×7	$+\infty$	49	33	16	6970	2882	4088	0	4088	23	250	14.8
10×10	1	100	67	33	15247	2751	2230	10266	1487	33	340	45.9
10×10	2	100	67	33	15247	5194	5055	4998	3312	63	91	358.0
10×10	$\sqrt{8}$	100	67	33	15247	6110	7274	1863	5423	25	169	86.8
10×10	3	100	67	33	15247	6503	7635	1109	6590	23	536	87.0
10×10	$\sqrt{13}$	100	67	33	15247	6839	8317	91	7592	25	299	123.2
10×10	4	100	67	33	15247	6899	8348	0	7924	26	283	171.8
10×10	$+\infty$	100	67	33	15247	6899	8348	0	8348	33	484	239.1
12×12	1	144	96	48	22977	2863	3323	16791	1143	92	462	532.8
12×12	2	144	96	48	22977	5907	6349	10721	3754	49	425	425.7
12×12	$\sqrt{8}$	144	96	48	22977	7712	8488	6777	5762	45	529	781.8
12×12	3	144	96	48	22977	9511	10135	3331	7257	16	153	195.1
12×12	$\sqrt{13}$	144	96	48	22977	10033	11568	1376	9070	50	184	966.3
12×12	4	144	96	48	22977	10225	12700	52	11288	47	573	1548.3
12×12	$+\infty$	144	96	48	22977	10251	12726	0	12726	58	657	1980.1
15×15	1	225	150	75	31354	3226	1956	26172	759	74	288	833.0
15×15	2	225	150	75	31354	6966	7458	16930	2625	43	611	1478.5
15×15	$\sqrt{8}$	225	150	75	31354	10029	10323	11002	5603	24	115	694.9
15×15	3	225	150	75	31354	12511	11883	6960	7503	22	93	743.9
15×15	$\sqrt{13}$	225	150	75	31354	13848	13071	4435	8992	22	15	1179.4
15×15	4	225	150	75	31354	15365	15181	808	12076	34	10	1973.3
15×15	$+\infty$	225	150	75	31354	16128	15226	0	15226	31	64	2035.3
17×17	1	289	192	97	43429	3297	3224	36908	1325	162	2950	10054.4
17×17	2	289	192	97	43429	7039	6254	30136	3409	218	5660	75792.1
17×17	$\sqrt{8}$	289	192	97	43429	10114	11454	21861	6071	162	3278	57669.4
17×17	3	289	192	97	43429	13000	13378	17051	8071	72	1953	22797.4
17×17	$\sqrt{13}$	289	192	97	43429	15215	14919	13295	10865	76	681	20989.0
17×17	4	289	192	97	43429	17392	18250	7787	13636	50	400	11575.2
17×17	$+\infty$	289	192	97	43429	20671	22758	0	22758	27	37	3436.5

Table 3: Statistics of our method for $B_l = B_f = 35$

Grid size	δ_j	$\|J\|$	$\|L\|$	$\|F\|$	Sum of demands	#Leader demand	#Follower demand	#Lost demand	#Stolen demand	#Cuts Total	#B&B Nodes	#Time(s) Total
5×5	1	25	16	9	3429	1321	1756	352	1388	7	39	0.6
5×5	2	25	16	9	3429	1405	2024	0	1893	7	14	0.6
5×5	$\sqrt{8}$	25	16	9	3429	1405	2024	0	2024	12	14	0.8
5×5	3	25	16	9	3429	1405	2024	0	2024	8	8	0.7
5×5	$\sqrt{13}$	25	16	9	3429	1405	2024	0	2024	9	8	0.8
5×5	4	25	16	9	3429	1405	2024	0	2024	9	13	0.7
5×5	$+\infty$	25	16	9	3429	1405	2024	0	2024	9	13	0.8
7×7	1	49	33	16	6970	2378	2568	2024	1311	15	81	3.3
7×7	2	49	33	16	6970	2893	3819	258	2954	20	60	10.1
7×7	$\sqrt{8}$	49	33	16	6970	2947	4023	0	3650	17	90	7.0
7×7	3	49	33	16	6970	2947	4023	0	3839	17	67	5.9
7×7	$\sqrt{13}$	49	33	16	6970	2947	4023	0	4023	16	98	6.5
7×7	4	49	33	16	6970	2947	4023	0	4023	17	51	6.9
7×7	$+\infty$	49	33	16	6970	2947	4023	0	4023	21	98	8.0
10×10	1	100	67	33	15247	3397	3995	7855	1937	61	756	124.6
10×10	2	100	67	33	15247	6010	6678	2559	4612	23	747	70.2
10×10	$\sqrt{8}$	100	67	33	15247	6878	8369	0	6928	39	203	102.0
10×10	3	100	67	33	15247	6884	8272	91	7051	49	301	203.0
10×10	$\sqrt{13}$	100	67	33	15247	6884	8363	0	7659	37	1515	161.5
10×10	4	100	67	33	15247	6884	8363	0	8175	54	1497	302.1
10×10	$+\infty$	100	67	33	15247	6884	8363	0	8363	61	1394	349.8
12×12	1	144	96	48	22977	3716	4673	14588	1847	330	1823	1909.0
12×12	2	144	96	48	22977	7451	9096	6430	4601	57	1513	861.5
12×12	$\sqrt{8}$	144	96	48	22977	8925	11053	2999	8980	77	835	1079.2
12×12	3	144	96	48	22977	10202	11607	1168	9866	17	94	114.8
12×12	$\sqrt{13}$	144	96	48	22977	10465	12395	117	11349	24	134	411.0
12×12	4	144	96	48	22977	10465	12512	0	12339	27	261	399.8
12×12	$+\infty$	144	96	48	22977	10465	12512	0	12512	47	515	1028.5
15×15	1	225	150	75	31354	4317	3021	24016	1160	92	1382	1245.3
15×15	2	225	150	75	31354	8696	7337	15321	4302	88	1509	6604.6
15×15	$\sqrt{8}$	225	150	75	31354	11768	11856	7730	8441	49	1155	3156.9
15×15	3	225	150	75	31354	13822	14783	2749	10854	32	412	1761.7
15×15	$\sqrt{13}$	225	150	75	31354	14464	15894	996	12922	45	269	2730.0
15×15	4	225	150	75	31354	15026	15538	790	14341	58	1217	5973.6
15×15	$+\infty$	225	150	75	31354	15374	15980	0	15980	75	340	8364.7

For all the types of budgets tested, the total demand served by the leader and the follower are similar for instances with large grids (12×12,15×15 and 17×17) leading us to think that for larger grid sizes, it also would happen.

Regarding the computational time, we observe that instances with grid size up to 12×12, it was necessary less than 1 hour to solve. On the other hand larger instances are harder for our method. For example, it was necessary almost 76000 seconds to solve the instance with $B_l = B_f = 25$, grid size 17×17 and $\delta_j = 2$. For this reason, we did not test instances with $B_l = B_f = 35$ and grid size 17×17.

5 Conclusions

In this paper, we presented a branch-and-cut algorithm for the discrete WCBC centroid problem using a new MIP formulation with polynomially many variables and exponentially many constraints. We also presented exhaustive experiments that proved the efficiency and robustness of our method. As future researches, we intend to generalize the others two cases proposed in Plastria and Vanhaverbeke (2008). The first one is when the leader wants to minimize it regret while in the second one, both the leader and the follower want to maximize their own market share.

Acknowledgements MCR received support from CAPES.

References

Eiselt, H. and Laporte, G. (1989), Competitive spatial models. *European Journal of Operational Research*, v. 39, n. 3, p. 231–242.

Eiselt, H. and Laporte, G. (1997), Sequential location problems. *European Journal of Operational Research*, v. 96, n. 2, p. 217–231.

Eiselt, H., Laporte, G. and Thisse, J. (1993), Competitive location models: A framework and bibliography. *Transportation Science*, v. 27, n. 1, p. 44.

Friesz, T., Miller, T. and Tobin, R. (1988), Competitive network facility location models: a survey. *Papers in Regional Science*, v. 65, n. 1, p. 47–57.

Hakimi, S. (1983), On locating new facilities in a competitive environment. *European Journal of Operational Research*, v. 12, n. 1, p. 29–35.

Kress, D. and Pesch, E. (2011), Sequential competitive location on networks. *European Journal of Operational Research*.

Noltemeier, H., Spoerhase, J. and Wirth, H. (2007), Multiple voting location and single voting location on trees. *European Journal of Operational Research*, v. 181, n. 2, p. 654–667.

Plastria, F. and Vanhaverbeke, L. (2008), Discrete models for competitive location with foresight. *Computers & Operations Research*, v. 35, n. 3, p. 683–700.

Roboredo, M. and Pessoa, A. (2011), The discrete one-roud voronoi game: Optimizing the first player strategy by integer programming. *Simpósio Brasileiro de Pesquisa Operacional*.