Integer programming formulations for three sequential discrete competitive location problems with foresight

José Gentile, Artur Alves Pessoa, Michael Poss, Marcos Costa Roboredo

Integer programming formulations for three sequential discrete competitive location problems with foresight

Jose Gentile

Production Engineering Department - Fluminense Federal University, Rua Passo da Pátria 156, 24210-240, Niterói, RJ, Brasil

Artur Alves Pessoa

Production Engineering Department - Fluminense Federal University, Rua Passo da Pátria 156, 24210-240, Niterói, RJ, Brasil

Marcos Costa Roboredo

Production Engineering Department - Fluminense Federal University, Rua Passo da Pátria 156, 24210-240, Niterói, RJ, Brasil

Michael Poss

UMR CNRS 5506 LIRMM, Université de Montpellier 2, 161 rue Ada, 34392 Montpellier Cedex 5, France.

Abstract

We deal with three competitive location problems based on the classical Maximal Covering Location Problem. The environment of these problems consists of an open market with two non-cooperative firms (leader and follower), several customers and locations where facilities can be located. In order to capture the demand of the customers, the leader enters the market by locating a set of facilities knowing the potential locations where the follower can locate her facilities after the leader's decision. We consider here three pairs of objective functions for the leader/follower previously studied in the literature: maximizing/minimizing the demand captured by the leader, minimizing/maximizing the regret of the leader, maximizing the demand

Email addresses: `josegentile@terra.com.br` (Jose Gentile), `artur@producao.uff.br` (Artur Alves Pessoa), `marcos.producao.uff@gmail.com` (Marcos Costa Roboredo), `michael.poss@lirmm.fr` (Michael Poss)

captured by each firm (also known as Stackelberg). For each model, we propose an linear integer programming formulation with a polynomial number of variables and an exponential number of constraints. The formulations are solved by branch-and-cut algorithms where the constraints are generated on demand by solving appropriated separation problems. We report extensive computational experiments realized on instances inspired by those from the literature, comparing our algorithms with the exact and heuristic algorithms previously published for these problems.

Keywords: Maximal Covering Location Problem, Competitive Location, Integer Programming, Stackelberg problem

---

1. Introduction

Location decision is a crucial issue in the strategic planning of a firm. A large part of research about location theory does not consider a potential competitor offering the same service or goods. In many real situations, it is vital to consider competition in the location decisions which leads to the competitive location problems (CLP). In these problems, two or more non-cooperative firms compete to capture customers from a given market. The demand of each customer is partially or totally served by the facilities located by the firms according to some given rule. Various types of CLP have been studied, which are often classified according to the structure of the decision space (discrete or continuous) and the temporal relations among the decisions (sequential or simultaneous). Different objective functions can be pursued by each firm when they decide where to locate its facilities. For example, they might seek to maximize their own market share, their own profit, their own number of customers served, among others. A review of competitive location problems can be found in Eiselt and Laporte (1989); Friesz et al. (1988); Eiselt et al. (1993); Eiselt and Laporte (1997); Kress and Pesch (2012).

This paper deals with three CLP variants that adds competition to the classical maximal covering location problem (MCLP) proposed by Church and Velle (1974). In MCLP, a firm should decide where to locate its facilities. After this decision, the demand of each customer in this market is said covered if there is a facility located by the firm within a threshold distance from the customer. In this context, the firm wishes to maximize the total covered demand.

In order to add competition to the MCLP, we consider two non-cooperative

firms named leader and follower. The leader enters the market by locating her facilities knowing that the follower will react by doing the same. After the location decisions, each customer demand is covered by the closest facility that is located within a threshold distance from the customer. We consider three pair of objective functions for the leader and the follower, following Plastria and Vanhaverbeke (2008): the leader maximizes her covered demand and the follower minimize that value (worst case criterion), the leader minimizes her regret while the follower maximizes that value (regret criterion), and both the leader and the follower maximize their respective covered demands (stackelberg criterion). We refer to those three problems as WC-CMCLP, RE-CMCLP and ST-CMCLP, respectively.

The WC-CMCLP and the RE-CMCLP were proposed by Serra et al. (1996) where the authors proposed an Integer Linear Programming (ILP) formulation for each problem considering a small number of possibilities for the follower's responses. The ST-CMCLP was proposed by Serra and ReVelle (1994) where a simple heuristic was proposed for the problem. The three problems were revisited in Plastria and Vanhaverbeke (2008) where a MILP formulation was proposed for each problem for the particular case where the follower is restricted to locating a unique facility. Recently, Seyhan (2012) proposed a heuristic for the ST-CMCLP for the case where the decision space of the follower is constrained by linear constraints, refereed to as the general case hereafter.

Contributions. To the best of our knowledge, no exact algorithm had been proposed for any of the three problems in the general case, prior to the current manuscript. Hence, our main contributions lie in presenting MILP-based algorithms that can solve these problems for the general case. Specifically, we propose for each variant a MILP formulation with a polynomial number of variables and an exponential number of constraints. The constraints are then generated on demand in the course of branch-and-cut algorithms by solving MILP separation problems. While the MILP formulations derived for WC-CMCLP extend rather naturally the previous work from Roboredo and Pessoa (2013), those presented for RE-CMCLP and ST-CMCLP require new insights on the problems studied. We provide numerical experiments assessing the efficiency of our algorithms on instances generated similarly to the procedure described in Plastria and Vanhaverbeke (2008). We also compare our method with the formulations proposed in Plastria and Vanhaverbeke (2008) for the particular case where the follower locates a single facility and

we also compare our algorithm for the general case of ST-CMCLP with the heuristic proposed in Seyhan (2012).

Structure of the paper. We present in the next section a formal definition of the three variants considered in the paper, including a numerical toy example. Section 3 complements the above literature review. Section 4 contains the main contributions of the paper, namely, the MILP formulations for the three problems as well as for the separation problems. The algorithmic details are provided in Section 5 and thorough numerical experiments are presented in Section 6, and in the supplementary material available online. The paper is finally concluded in Section 7.

2. The problems

We define next the three problems formally. We consider three sets: the set of customers, denoted J, the set of leader potential facilities, denoted L, and the set of follower potential facilities, denoted F. Let $x^l \in \{0,1\}^{|L|}$ and $x^f \in \{0,1\}^{|F|}$ be binary decision variables indicating which facilities are located by the leader and the follower, respectively. We restrain the sets of facilities that can be located by the leader and the follower by $\mathcal{X}^l \subseteq \{0,1\}^{|L|}$ and $\mathcal{X}^f \subseteq \{0,1\}^{|F|}$, respectively. Sets $\mathcal{X}^l$ and $\mathcal{X}^f$ can, for instance, be defined by one ore more budget constraints. Specifically, locating facility $i \in L \cup F$ would incur a cost of κ_i and the leader and the follower would dispose of budgets B^l and B^f, respectively. We could also suppose that $\mathcal{X}^l$ (resp. $\mathcal{X}^f$) contains incompatibility constraints, forbidding the leader (resp. the follower) to simultaneously locate pairs of given facilities. For each pair $(i,j) \in (L \cup F) \times J$, we denote by d_{ij} the distance between facility i and customer j. Finally, each facility $i \in L \cup F$ has a radius of δ_i that restrains the customers that it can attend. Each customer's demand w_j is totally served by the closest facility $i \in L \cup F$ that satisfies $\delta_i \geq d_{ij}$. Ties are broken in favor of the follower's facilities, and ties between facilities of the same firm are broken arbitrarily.

The difference among the problems lie in the objective functions of the leader and the follower. In the WC-CMCLP, the leader aims to maximize its market share while the follower aims to minimize the leader market share. In the RE-CMCLP, the leader aims to minimize its maximum regret while the follower aims to maximize the leader's maximum regret. The regret of the leader is given by the difference between the maximum demand served by

4

the leader, if the follower's decision were previously known, and the demand actually served by this firm. Finally, in the ST-CMCLP both the leader and the follower aim to maximize their own market share.

The three problems can be cast as special cases of the following Bilevel Integer Programming Problem (BIP). Let us denote the objective functions of the leader and the follower by $g^l(x^l, x^f)$ and $g^f(x^l, x^f)$, respectively.

$$\max \ g^l(x^l, x^f) \tag{1}$$
$$\text{s.t. } x^l \in \mathcal{X}^l \tag{2}$$
$$x^f \in \arg\max \ g^f(x^l, x^f) \tag{3}$$
$$\text{s.t. } x^f \in \mathcal{X}^f. \tag{4}$$

To ease the definition of the functions g^l and g^f used in each of the three variants studied in the paper, we introduce sets $J^l(x^l, x^f) \subseteq J$ and $J^f(x^l, x^f) \subseteq J$ that denote the sets of customers that are served by the leader and the follower, for the locating (x^l, x^f), respectively. We consider the following specific functions g^l and g^f:

WC-CMCLP: $g^l(x^l, x^f) = \sum_{j \in J^l(x^l, x^f)} w_j$ and $g^f(x^l, x^f) = -g^l(x^l, x^f)$.

RE-CMCLP: Let $x^*(x^f) \subseteq \{0, 1\}^{|L|}$ denote the leader best response knowing that the follower will locate the facilities indicated by x^f. Hence, the regret of the pair (x^l, x^f) is given by

$$Regret(x^l, x^f) = \sum_{j \in J^l(x^*(x^f), x^f)} w_j - \sum_{j \in J^l(x^l, x^f)} w_j, \tag{5}$$

yielding the objective functions $g^l(x^l, x^f) = -Regret(x^l, x^f)$ and $g^f(x^l, x^f) = -g^l(x^l, x^f)$.

ST-CMCLP: $g^l(x^l, x^f) = \sum_{j \in J^l(x^l, x^f)} w_j$ and $g^f(x^l, x^f) = \sum_{j \in J^f(x^l, x^f)} w_j$.

Let us make some observations on the structure of the three variants of CMCLP defined above. First, when each pre-specified distances δ_i is large enough to cover all customers, we have that $J^l \cup J^f = J$ so that the WC-CMCLP and ST-CMCLP are equivalent. In general however, we have that $J^l \cup J^f \subset J$ and these problems are different, see the toy examples below. Second, for WC-CMCLP and RE-CMCLP, we have $g^f(x^l, x^f) = -g^l(x^l, x^f)$ so that both problems can be reformulated as

$$\max_{x^l \in \mathcal{X}^l} \ \min_{x^f \in \mathcal{X}^f} \ g^l(x^l, x^f). \tag{6}$$

Table 1: Total demand served by the leader and the follower for each pair of strategies of the leader and the follower.

		Follower		
		F1	F2	F3
Leader	L1	$(5,4)$	$(4,5)$	$(5,6)$
	L2	$(3,7)$	$(7,2)$	$(5,6)$

Reformulation (6) better sheds light on the names *worst-case* and *regret* used to refer to these problems. Notice also that while (6) resembles robust $\min \max$ optimization, the integrality restriction in $\mathcal{X}^f$ make the problem fundamentally different from robust optimization wherein the inner optimization problem is defined over real variables. The integrality of the optimization vector x^f also implies that the algorithms described in this paper, based on constraint generation with MILP separation problems, are not related to Benders' decomposition, which relies on LP separation problems. Differently from Benders' decomposition, our approaches crucially rely on the fact that the feasibility set $\mathcal{X}^l$ (resp. $\mathcal{X}^f$) does not depend on x^f (resp. x^l), which allows us to propose in Section 4 models containing specific constraints for each vector in $\mathcal{X}^f$.

We illustrate below the above formal definition on a toy example in which $J^l \cup J^f \subset J$. The example contains twelve customers ($|J| = 12$), each having a demand of one unit, two potential facilities for the leader ($L = \{L_1, L_2\}$) and three potential facilities for the follower ($F = \{F_1, F_2, F_3\}$). Sets $\mathcal{X}^l$ and $\mathcal{X}^f$ are characterized by the fixed costs κ_i and the budgets B^l and B^f are unitary. In other words, both the leader and follower can locate a unique facility. The distances d_{ij} between the potential facilities and the customers, the radius δ_i and the demands w_j are given by Table 1 implicitly. That table shows the total demand served by the leader and the follower in this order for each possible pair of strategies of the leader and the follower.

According to table 1 if the leader locates the facility L_2 and the follower locates F_1 then the leader serves a total demand of three units while the follower serves a total of seven units. Note that in the worst case, the leader serves at least four and three units when she locates L_1 and L_2, respectively. Hence, when we consider the WC-CMCLP, the leader locates the facility associated to the maximum of these two numbers, L_1.

Table 2: Regret of the leader for each pair of strategies of the leader and the follower.

		Follower		
		F1	F2	F3
Leader	L1	0	3	0
	L2	2	0	0

In the ST-CMCLP the leader's location decision is made in the following way. On the one hand, if the leader locates L_1 then the follower locates F_3, which is the facility that maximizes her market share. Hence, the leader would serve four units. On the other hand, if the leader locates L_2 then the follower locates F_1. Consequently, the leader would serve three units. Observing the two cases, the leader locates L_1. Note that, while the leader locates L_1 in the optimal solutions to WC-CMCLP and ST-CMLP, the decisions of the follower are different, locating F_2 in the former and F_3 in the latter.

To identify which strategy would be used by the leader in the RE-CMCLP, we first construct Table 2 containing the values of *Regret* for each pair of strategies of the leader and the follower. The table is filled in the following way. Suppose that the leader locates the facility L_2 and the follower locates F_1. Hence, the leader serves a total demand of three units but she could have served five units if she had located L_1. Hence the leader's regret is $5 - 3 = 2$ units for the pair (L_2, F_1). The purpose of the leader is to minimize her maximal regret. If the leader locates L_1, the greatest regret is 3 (follower locates F_2). If the leader locates L_2, the greatest regret is 2 (follower locates F_1). Therefore the leader should locate L_2, remaining with a regret of at most 2.

3. Review of the literature

The WC-CMCLP, the RE-CMCLP and the ST-CMCLP can be seen as special Bilevel Integer Programming Problems where the first level decision consists of choosing where the leader locates her facilities while the second level is associated with the follower's location decisions. Yet, the WC-CMCLP and the RE-CMCLP are very special cases of BIP since they can be rewritten as min max optimization problems. The literature is rather

scarce when it comes to exact methods for general BIPs. Branch-and-bound algorithms for linear BIP were proposed by Moore and Bard (1990) and Bard and Moore (1992). However, these are able to solve only instances with up to 10 general integer variables and 35 binary variables for the first level problem. Other approaches are based on specific characteristics of the optimization problem under study such as a decomposition method based on super valid inequalities, see Israeli and Wood (2002); O'Hanley and Church (2011), cutting plane algorithms Taşkın et al. (2009), among others. Recently, Fischetti et al. (2016) have proposed generic algorithms to solve larger BIP to optimality, relying on intersection cuts.

Coming back to the CMLP, Plastria and Vanhaverbeke (2008) revisited the three aforementioned variants and proposed a MILP formulation for each of them considering the particular case where the follower locates a unique facility. In order to show the robustness of the formulations, the authors presented several numerical results for randomly generated instances. Although the results are able to solve instance of moderate size efficiently, it does not seem possible to extend their formulations to the more general problems considered herein. Recently, Seyhan (2012) considered the general case of the ST-CMCLP and propose a greedy heuristic for the problem. That heuristic is based on a MILP formulation where a set of variables and linear constraints are used to compute the follower response in a greedy heuristic way.

A closely related problem is the discrete $(r|p)$-centroid problem, proposed by Hakimi (1983). In the $(r|p)$-centroid problem, the coverage radii are infinite, $\delta_i = +\infty$, and all fixed costs κ_i are unitary. In other words, each customer is covered by one of the facilities and there is a fixed number of facilities to be located by the leader and the follower. Since Noltemeier et al. (2007) proved that the $(r|p)$-centroid problem is Σ_2^p-hard, the previous observation implies that both the WC-CMCLP and the ST-CMCLP are Σ_2^p-hard. Moreover, some techniques used for the $(r|p)$-centroid problem can be adapted for the WC-CMCLP and ST-CMCLP. Specifically, the algorithm used in this paper for the WC-CMCLP is an adaptation of the branch-and-cut algorithm proposed by Roboredo and Pessoa (2013) for the $(r|p)$-centroid problem.

4. MILP Formulations for the problems

In this section, we present for each of the three problems a MILP formulation with a polynomial number of variables and a exponential number

8

of constraints together with a compact MILP formulation for separating the exponential number of constraints.

We use the following notations in the formulations: let $\mathcal{S}^f$ be the set that contains the ground sets of all vectors in $\mathcal{X}^f$, that is, the set of feasible strategies for the follower. Formally, each $S_0 \in \mathcal{S}^f$ represents a subset of F such that the vector x^f defined by $x_i^f = 1$ for $i \in S_0$ and $x_i^f = 0$ otherwise belongs to $\mathcal{X}^l$. Furthermore, for each customer $j \in J$ and for each $S_0 \in \mathcal{S}^f$, we define $d(S_0, j) = \min \{d_{kj} | k \in S_0\}$. Finally, for each customer $j \in J$, the sets $L'(j) = \{i \in L | d_{ij} \leq \delta_i\}$ and $F'(j) = \{i \in F | d_{ij} \leq \delta_i\}$ contain the facilities of the leader and the follower that can potentially cover customer j.

4.1. MILP formulation for the WC-CMCLP

Let z_{wc} be a non-negative continuous variable indicating the total demand served by the leader after the follower's response. For each $j \in J$ and $i \in L$, the binary first level variable y_{ij}^l is equal one if and only if the facility i is located by the leader to try to serve customer j. The formulation follows.

$$\max \; z_{wc} \tag{7}$$

$$\text{s.t.} \; y_{ij}^l \leq x_i^l, \qquad\qquad \forall j \in J, \forall i \in L \tag{8}$$

$$\sum_{i \in L} y_{ij}^l = 1, \qquad\qquad \forall j \in J \tag{9}$$

$$z_{wc} \leq \sum_{j \in J} \sum_{\substack{i \in L'(j) \\ d_{ij} < d(S_0, j)}} w_j y_{ij}^l \qquad \forall S_0 \in \mathcal{S}^f \tag{10}$$

$$y_{ij}^l \in \{0, 1\}, \qquad\qquad \forall i \in L, \; \forall j \in J \tag{11}$$

$$z_{wc} \geq 0, \tag{12}$$

$$x^l \in \mathcal{X}^l. \tag{13}$$

The objective function (7) maximizes the total demand served by the leader after the follower's response. Constraints (8) ensure the consistency between variables x^l and y^l. Constraints (9) indicate that for each customer j, the leader locates exactly one facility to try to serve customer j. To explain constraints (10), consider a given set $S_0 \in \mathcal{S}^f$ and a customer $j \in J$. Then,

$$\sum_{\substack{i \in L'(j) \\ d_{ij} < d(S_0, j)}} w_j y_{ij}^l$$

9

is equal to w_j only if the facility chosen by the leader is closer than the closest admissible facility of the follower in strategy S_0. Hence, summing up over j yields the revenue of the leader for the follower strategy S_0 and writing the constraint for each $S_0 \in \mathcal{S}^f$ yields the worst-case among all strategies possible for the follower. Note that (10) is composed of an exponential number of constraints. Hence, it is necessary to solve the separation problem associated to (10) in order to include in the formulation only the necessary constraints.

Given a relaxed leader solution $(\overline{z}_{wc}, \overline{x}^l, \overline{y}^l) \in \mathbb{R}^+ \times [0,1]^{|I|} \times [0,1]^{|I| \times |J|}$ possibly violating some of the constraints in (10), the separation problem for the constraints (10) consists of finding the follower's strategy that minimizes the total demand served by the leader. That strategy can be found by solving the integer programming model described below.

Recall that, for each $k \in F$, x_i^f is a binary variable indicating whether the follower locates facility k. Similarly, for each $j \in J$ and $k \in F$, let the binary variable y_{kj}^f indicate if the facility k is the closest facility of the follower to customer j. The formulation follows.

$$\min \sum_{j \in J} \sum_{k \in F'(j)} \left(w_j \sum_{\substack{i \in L'(j) \\ d_{ij} < d_{kj}}} \overline{y}_{ij}^l \right) y_{kj}^f \tag{14}$$

$$\text{s.t. } y_{kj}^f \leq x_k^f, \qquad\qquad \forall j \in J, \forall k \in F \tag{15}$$

$$\sum_{k \in F} y_{kj}^f = 1, \qquad\qquad \forall j \in J \tag{16}$$

$$y_{kj}^f \in \{0,1\}, \qquad\qquad \forall k \in F, \forall j \in J \tag{17}$$

$$x^f \in \mathcal{X}^f. \tag{18}$$

The objective function (14) minimizes the total demand served by the leader. Constraints (15) ensure the consistency between the variables y^f and x^f. Constraints (16) indicate that for each customer j, the follower locates exactly one facility to try to serve customer j.

The formulation (7) – (12) for the leader problem and the formulation (14) – (18) for the separation problem were adapted from the formulations presented by Roboredo and Pessoa (2013) for the $(r|p)$-centroid problem.

4.2. MILP formulation for the RE-CMCLP

We present below a MILP formulation for the RE-CMCLP. The variables x^f, y^f used below have the same meaning as in (7) – (12). Let z_{re} be a non-negative continuous variable indicating the worst regret for the leader. For each $S_0 \in \mathcal{S}^f$ we define $z^*(S_0)$ as the largest demand served by the leader if she knows that the follower strategy is S_0. Recalling the definition (5) from Section 2, we can define $z^*(S_0)$ formally as

$$z^*(S_0) = \sum_{j \in J^l(x^*(x_0^f), x_0^f)} w_j,$$

where $x_{0i}^f = 1$ for $i \in S_0$ and $x_{0i}^f = 0$ for $i \in F \setminus S_0$. The formulation follows.

$$\min z_{re} \tag{19}$$
$$\text{s.t. } y_{ij}^l \leq x_i^l, \qquad\qquad\qquad \forall j \in J, \forall i \in L \tag{20}$$
$$\sum_{i \in L} y_{ij}^l = 1, \qquad\qquad\qquad \forall j \in J \tag{21}$$
$$z_{re} \geq z^*(S_0) - \sum_{j \in J} \sum_{\substack{i \in L'(j) \\ d_{ij} < d(S_0, j)}} w_j y_{ij}^l \qquad \forall S_0 \in \mathcal{S}^f \tag{22}$$
$$y_{ij}^l \in \{0, 1\}, \qquad\qquad\qquad \forall i \in L, \forall j \in J \tag{23}$$
$$z_{re} \geq 0, \tag{24}$$
$$x^l \in \mathcal{X}^l. \tag{25}$$

The above formulation is similar to the formulation provided in the previous section, with the following differences. First, the objective function (19) minimizes the worst regret z_{re}, while objective function (7) maximizes the worst revenue. Second, and more importantly, constraints (22) involve a constant term, $z^*(S_0)$, that is not present in (10). Again, the number of constraints in (22) is typically exponential so that the constraints are generated on demand in a branch-and-cut algorithm. The separation problem associated to (22) is defined in the following way. Given a relaxed leader solution $(\overline{z}_{re}, \overline{x}^l, \overline{y}^l) \in \mathbb{R}^+ \times [0, 1]^{|I|} \times [0, 1]^{|I| \times |J|}$ possibly violating some of the constraints in (22), the separation problem for the constraints (22) consists of finding the feasible follower strategy that maximizes the regret of the leader. That strategy can be found by solving the a MILP model introduced next.

Recall that the regret of the leader is given by a difference of two terms where the first one is associated to the best leader strategy given a feasible follower strategy. From now on, we refer to that leader as the *virtual leader*. According to the notations introduced before, for each $i \in L$, the binary variable x_i^* is equal one if and only if the virtual leader locates facility i. For each $j \in J$ and $i \in L$, the binary variable y_{ij}^* is equal one if and only if facility i is used by the virtual leader to serve customer j. Before proceeding with the definitions used in the formulation, let us make a simple, yet crucial, observation. In the separation problem, variables x^* and x^f purse the same objective, that is, maximizing the regret. Hence, although the separation problem involves decision variables for the (virtual) leader and the follower, it is not a BIP. As we show below, the separation problem can in fact be modeled quite naturally as a MILP with polynomial numbers of variables and constraints.

For each $j \in J$ and $k \in F$, the binary variable v_{kj}^f is equal to is equal to 1 if and only if the follower does not locate facility k ($x_k^f = 0$) and there is no facility located by the follower closer to customer j than k ($x_{k'}^f = 0, \forall k' \in F | d_{k'j} < d_{kj}$), otherwise the variable is equal to 0. Finally the separation model also relies on variables x^f and y^f already defined for the formulation (7) – (12). In the formulation below, we also use the constant $\theta(k, j)$ to denote the k-th potential facility of the follower closer to customer j, for each customer $j \in J$ and each follower potential facility $k \in F$.

$$\max \sum_{j \in J} \sum_{i \in L'(j)} w_j y_{ij}^* - \sum_{j \in J} \sum_{k \in F'(j)} \left(w_j \sum_{i \in L'(j) | d_{ij} < d_{kj}} \overline{y}_{ij}^l \right) y_{kj}^f \qquad (26)$$

$$\text{s.t.} \quad y_{kj}^f \le x_k^f, \qquad \qquad \forall j \in J, \forall k \in F$$

$$\tag{27}$$

$$\sum_{k \in F} y_{kj}^f = 1, \qquad \qquad \forall j \in J$$

$$\tag{28}$$

$$y_{ij}^* \le x_i^*, \qquad \qquad \forall j \in J, \forall i \in L$$

$$\tag{29}$$

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

$$\tag{30}$$

$$y_{ij}^* + \sum_{k \in F'(j) | d_{kj} \le d_{ij}} y_{kj}^f \le 1, \qquad \qquad \forall j \in J, \forall i \in L'(j)$$

$$\tag{31}$$

$$v_{\theta(1,j)j}^f = 1 - x_{\theta(1,j)}^f, \qquad \qquad \forall j \in J$$

$$\tag{32}$$

$$v_{kj}^f \le 1 - x_k^f, \qquad \qquad \forall k \in F, \forall j \in J$$

$$\tag{33}$$

$$v_{\theta(k,j)j}^f \ge v_{\theta(k+1,j)j}^f, \qquad \qquad \forall k \in \{1, ..., |F|-1\}, \forall j \in J$$

$$\tag{34}$$

$$y_{\theta(1,j)j}^f = 1 - v_{\theta(1,j)j}^f, \qquad \qquad \forall j \in J$$

$$\tag{35}$$

$$y_{\theta(k,j)j}^f = v_{\theta(k-1,j)j}^f - v_{\theta(k,j)j}^f, \qquad \qquad \forall k \in \{2, ..., |F|\}, \forall j \in J$$

$$\tag{36}$$

$$y_{kj}^f \in \{0,1\}, \qquad \qquad \forall j \in J, \forall k \in F'(j)$$

$$\tag{37}$$

$$y_{kj}^* \in \{0,1\}, \qquad \qquad \forall j \in J, \forall i \in L'(j)$$

$$\tag{38}$$

$$x^f \in \mathcal{X}^f, \qquad \qquad \tag{39}$$

$$x^* \in \mathcal{X}^l. \qquad \qquad \tag{40}$$

The objective function (26) maximizes the leader regret. Constraints (31) ensure that if y_{ij}^* is equal 1, all facilities located by the follower are further

from j than facility i. Consider then constraints $(32) - (36)$ and suppose that these constraints are removed from the formulation. Then, the virtual leader and the follower can falsely increase the optimal solution cost by letting the follower try to capture the customer by a distant facility instead of assigning her closest facility. This increases the profit of the virtual leader and reduces the profit of the follower, thus falsely increasing the optimal solution cost.

Constraints (32) ensure that if the follower locates her closest facility from customer j, then the corresponding component of v is equal to 0. Otherwise, the component of v is equal to 1. Constraints (33) ensure that if the follower locates facility k, v_{kj}^f is equal to 0, for each customer j. Constraints (34) ensure that if the variable v_{kj}^f corresponding to the k-th closest facility from customer j is equal to 0, then the subsequent (from $k+1$ until $|F|$) variables are also equal to 0. Constraints (35) and (36) ensure the consistency between the variables y^f and v. Constraints (35) ensure that if follower locates the closest facility from customer j, then y_{kj}^f is equal to 1 and v_{kj}^f is equal to 0. Otherwise, y_{kj}^f is equal to 0 and v_{kj}^f is equal to 1. Constraints (36) ensure that if the k-th potential facility closest from customer j is the facility located by the follower closest to customer j, then $y_{\theta(k,j)j}^f$ is equal to 1, $v_{\theta(k-1,j)j}^f$ is equal to 1 and $v_f\theta(k,j)j$ is equal to 0.

It is worth noting that constraints $(32) - (36)$ can be relocated by

$$y_{kj}^f + x_{k'}^f \leq 1, \qquad \forall j \in J, \forall k \in F'(j),\ \forall k' \in F'(j) | d_{k'j} < d_{kj}, \qquad (41)$$

eliminating the variables v from the formulation. However, despite having a simpler description than the previous ones, constraints (41) corresponds to $O(|J||F|^2)$ inequalities. Unreported results showed that constraints $(32) - (36)$ are numerically more efficient than constraints (41) for the large instances.

4.3. MILP formulation for the ST-CMCLP

We present next a MILP formulation for the ST-CMCLP. Differently from the formulations proposed for the previous two problems, the formulation proposed here for the ST-CMCLP contains variables related to both the leader and the follower strategies. As before, these variables are respectively denoted by x^l and y^l, and x^f and y^f, respectively. In addition, we introduce binary variable z_j^l (resp. z_j^f) that is equal to one if and only if costumer j is served by the leader (resp. follower). We also define the vector of optimization variables v^l and the constants λ, which are the counterpart of

14

v and θ for the leader's strategy. The variables ensure that y^l is set to 1 for the closest facility. The formulation follows.

$$\max \ \sum_{j \in J} w_j z_j^l \tag{42}$$

subject to:

$$y_{ij}^l \leq x_i^l \qquad\qquad\qquad \forall j \in J, \forall i \in L \tag{43}$$

$$y_{kj}^f \leq x_k^f \qquad\qquad\qquad \forall j \in J, \forall k \in F \tag{44}$$

$$\sum_{i \in L} y_{ij}^l = 1 \qquad\qquad\qquad \forall j \in J \tag{45}$$

$$\sum_{k \in F} y_{kj}^f = 1 \qquad\qquad\qquad \forall j \in J \tag{46}$$

$$z_j^l + z_j^f \leq 1 \qquad\qquad\qquad \forall j \in J \tag{47}$$

$$z_j^l \leq \sum_{i \in L'(j)} y_{ij}^l \qquad\qquad\qquad \forall j \in J \tag{48}$$

$$z_j^f \leq \sum_{k \in F'(j)} y_{kj}^f \qquad\qquad\qquad \forall j \in J \tag{49}$$

$$z_j^l \geq \sum_{k \in L'(j)|d_{kj} \leq d_{ij}} y_{kj}^l - \sum_{k \in F'(j)|d_{kj} \leq d_{ij}} y_{kj}^f \qquad \forall j \in J, \forall i \in L'(j) \tag{50}$$

$$z_j^f \geq \sum_{k \in F'(j)|d_{kj} \leq d_{ij}} y_{kj}^f - \sum_{k \in L'(j)|d_{kj} \leq d_{ij}} y_{kj}^l \qquad \forall j \in J, \forall i \in F'(j) \tag{51}$$

$$v_{\lambda(1,j)j}^l = 1 - x_{\lambda(1,j)}^l \qquad\qquad\qquad \forall j \in J \tag{52}$$

$$v_{ij}^l \leq 1 - x_i^l \qquad\qquad\qquad \forall j \in J, \forall i \in L \tag{53}$$

$$v_{\lambda(i,j)j}^l \geq v_{\lambda(i+1,j)j}^l \qquad\qquad\qquad \forall j \in J, \forall i = 1, ..., L-1 \tag{54}$$

$$y_{\lambda(1,j)j}^l = 1 - v_{\lambda(1,j)j}^l \qquad\qquad\qquad \forall j \in J \tag{55}$$

$$y_{\lambda(i,j)j}^l = v_{\lambda(i-1,j)j}^l - v_{\lambda(i,j)j}^l \qquad\qquad\qquad \forall j \in J, \forall i = 2, ..., L \tag{56}$$

$$\sum_{j \in J} w_j z_j^f \geq \sum_{\substack{j \in J: \\ S_0 \cap F'(j) \neq \emptyset}} w_j \left(1 - \sum_{\substack{i \in L'(j): \\ d_{ij} < \min\{d_{kj} | k \in S_0\}}} y_{ij}^l \right) \qquad \forall S_0 \in \mathcal{S}^f \tag{57}$$

$$y_{ij}^l, y_{kj}^f, z_j^l, z_j^f \in \{0,1\} \qquad \forall j \in J, \forall i \in L, \forall k \in F, \tag{58}$$

$$x^l \in \mathcal{X}^l, \tag{59}$$

$$x^f \in \mathcal{X}^f. \tag{60}$$

The objective function (42) maximizes the leader market share. The constraints (43)–(46) are identical to those introduced in the previous formulations. Constraints (47) ensure that each costumer j be served either by the leader or the follower. Constraints (48) (resp. (49)) ensure that if a costumer is served by the leader (resp. follower), the nearest facility from this costumer located by the leader (resp. follower) is within the coverage radius. Constraints (50) ensure that if costumer j is not covered by the leader (resp. follower), the nearest facility from this costumer, located within coverage by the leader, is not closer to those located within coverage by the follower (resp. leader). Constraints (52) – (56) are the counterparts of constraints (32) – (36) for the leaders variables. As in the RE-CMCLP, these 5 group of constraints could be relocated by a single one which corresponds to $O(|J||F|^2)$ inequalities. Again, unreported experimental results showed that the use of constraints (52) – (56) obtained better results for large instances. There was also a quick explosion of the constraint number in the case where a single group of constraints is used.

Finally, constraints (57) ensure that the follower chooses her best location. In other words, the constraints ensure that, given the strategy taken by the leader, the follower will pick up the one which maximizes her covered demand. Note that the right-hand side of constraints (57) shows the demand served by the follower, given her strategy S_0 and the leader strategy defined by variables y^l. More explicitly, this demand is the difference between all demand that the follower could cover and the demand taken by the leader due to her closer facilities. There is a constraint (57) for each feasible strategy for the follower S_0. As this number of constraints tends to be exponential, the constraints are added on demand during a branch-and-cut algorithm. We propose below an exact ILP formulation problem to separate the cuts (57) in the line of those proposed in for the other two variants.

Given a relaxed leader solution $(\bar{x}^l, \bar{y}^l, \bar{z}^l) \in [0,1]^{|I|} \times [0,1]^{|I| \times |J|} \times [0,1]^{|J|}$

for the model (42) – (58) possibly violating some of the constraints (57), the separation problem amounts to find the follower strategy that maximizes the violation of the constraint, which is to maximize the right side of (57).

$$\max \sum_{j \in J} \sum_{k \in F} \left(w_j \sum_{i \in L'(j) | d_{kj} \leq d_{ij}} \overline{y}_{ij}^l \right) y_{kj}^f \tag{61}$$

$$\text{s.t. } y_{kj}^f \leq x_k^f, \qquad\qquad \forall j \in J, \forall k \in F \tag{62}$$

$$\sum_{k \in F} y_{kj}^f = 1, \qquad\qquad \forall j \in J \tag{63}$$

$$y_{kj}^f \in \{0, 1\}, \qquad\qquad \forall k \in F, \forall j \in J \tag{64}$$

$$x^f \in \mathcal{X}^f. \tag{65}$$

5. Details of the branch-and-cut algorithms

The exact algorithms for the WC-CMCLP, the RE-CMCLP and the ST-CMCLP are built on the top of the branch-and-cut algorithms that solve the models given by, respectively, (7) – (12), (19) – (25) and (42) – (58) using commercial solvers. Constraints (10), (22) and (57) are added through cut callbacks by solving, respectively, the separation problems (14) – (18), (26) – (40) and (61) – (64). Notice that our separation algorithms can be used to separate cuts over fractional solutions along the branch-and-bound tree. We did it only for the ST-CMCLP where it improved the results significantly. More specifically, we separate cuts over fractional solutions at the root node until the relative difference between the current relaxation and the best integer solution found so far drops below a given threshold ϵ. We also limited the number of separations over fractional solutions to 200.

For the three problems, we also used the following technique to obtain quickly feasible integer solutions. Every time we reach an integer branch-and-bound node that violates some of the missing constraints, the algorithm inserts the following incumbent solution: the leader variables are set to their values in the current node, while the follower variables are set to their values in the optimal solution of the separation problem. Note that the time spent by this strategy is negligible.

For the ST-CMCLP, we observed that it takes too much time to find good integer solutions even using the previous technique. Therefore, we propose

the following procedure able to transform fractional solutions into feasible integer ones. Whenever a fractional relaxed solution does not violate any missing inequality, we take the solution of the follower obtained from solving the separation problem and compute the leader strategy that maximizes her covered demand. This amounts to solve a MILP very close to formulation (61) – (64), exchanging the roles of the leader and the follower. In a second step, we find a new follower solution running model (61) – (64) based on the leader solution from the previous step. The strategies of the leader and follower found in the previous two steps form a feasible integer solution to the problem, which is fed to the algorithm. Note that the computational time of this procedure require solving two MILP formulations.

6. Computational Results

In this section, we present our computational results. The tests were coded in C++, using CPLEX 12.5.1, and the experiments were conducted on a computer with an Intel core i7-4790 3.60GHz CPU and 16 GB of memory running the windows 8 operating system. A single thread was used for the tests. All instances used in this paper are available in `www.logis.uff.br/ ~roboredo/instances`.

We tested our three methods on randomly instances generated similarly to Plastria and Vanhaverbeke (2008). Specifically, the coordinates of the customers and facilities are randomly distributed on square grids, where each grid cell has integer positive coordinates. There are not common potential facilities for the leader and for the follower ($L \cap F = \emptyset$). The potential facilities for the follower (set F) are those cells for which sum of the coordinates is a multiple of 3 while the potential facilities for the leader (set L) consists of all other cells, so approximately $|L| \approx 2|F|$. We considered the euclidean distances between customers and facilities. Customer demands w_j and fixed costs κ_i were integers uniformly generated in $[50, 250]$ and $[5, 10]$, respectively. The instance types and their characteristics are summarized in Table 3.

For each type of instance, we considered the following covering radii, identical for all facilities: 1, $\sqrt{2}$, 2, $\sqrt{5}$, $\sqrt{8}$, 3, $\sqrt{10}$, $\sqrt{13}$ and 4. Sets $\mathcal{X}^l$ and $\mathcal{X}^f$ are defined by binary restrictions and single budget constraints, defined by the budgets $B^l = B^f$ ranging in $\{15, 25, 35\}$.

In sections 6.1, 6.2 and 6.3 we respectively present average statistics considering all covering radii for the WC-CMCLP, the RE-CMCLP and the ST-

18

Table 3: Instance types and their characteristics

Type	Grid	$|J|$	$|L|$	$|F|$	$\sum w_j$
P5	5x5	25	16	9	3542
P7	7x7	49	33	16	6531
P10	10x10	100	67	33	15247
P12	12x12	144	96	48	22977
P15	15x15	225	150	75	31354
P17	17x17	289	192	97	43999
P20	20x20	400	134	266	61910

CMCLP. We provide detailed results in the supplementary material available online. Section 1 therein presents the comparison between our methods and the one proposed in Plastria and Vanhaverbeke (2008) for the particular case where the follower places just one facility. Section 2 therein presents detailed statistics of the branch-and-cut algorithms for each one of the instances tested. Finally,

6.1. Results for the WC-CMCLP

Table 4 presents several statistics of our method for the WC-CMCLP. Specifically, for each type of instance and each budget, the table shows average results considering all covering radii. The following headers are used for the columns. Column *Budget* indicates the budget of the firms. Column *Type* indicates the type of the instance (according Table 3). *Obj* indicates the total demand covered by the leader at the optimal solution, *Gap Root(%)* indicates the percentage of the relative difference between the best upper bound obtained at the root node and the optimal solution, *Time Root(s)* indicates the CPU times in seconds spent at the root node, *#Nodes* indicates the total number of nodes of the B&B tree, *#Sep* indicates the number of separation problems solved, *#Cuts* indicates the total number of cuts generated by separation problems, *#Time Sep(s)* and *Time total(s)* indicate, respectively, the total CPU time in seconds spent by the separation problems and the complete branch-and-cut algorithm. If the optimal solution was not found after 21600 seconds of execution we stopped the algorithm and report the best feasible solution found.

Observing Table 4 we note that the branch-and-cut algorithm found the

Table 4: WC-CMCLPC: average results considering all covering radii

Instance		Root node				B&B Tree			
Budget Type		Obj	Gap(%)	Time(s)	#Nodes	#Sep	#Cuts	Sep Time(s)	Total Time(s)
	P5	1815.56	0.47	0.25	0.22	9.11	5.44	0.17	0.24
	P7	2270.67	9.30	1.23	14.33	32.67	9.67	1.28	1.92
	P10	4697.44	6.08	7.78	19.44	39.78	13.44	7.25	10.48
15	P12	6516.44	9.00	7.04	29.33	57.00	16.00	13.42	20.72
	P15	8028.89	10.33	59.11	85.22	158.67	42.44	96.15	139.36
	P17	8374.44	21.65	24.95	104.67	244.11	86.33	411.76	590.15
	P20	8815.89	16.41	557.42	270.22	545.11	170.56	2124.70	2840.08
	P5	1406.89	5.89	0.52	16.33	37.11	9.44	0.37	0.54
	P7	2263.89	7.13	2.34	35.78	75.44	20.33	4.32	5.19
	P10	5229.56	12.12	14.90	181.11	300.67	61.00	49.06	60.91
25	P12	8042.22	5.04	13.28	65.78	111.11	20.44	12.23	20.97
	P15	10725.56	8.70	76.68	286.00	470.11	103.00	257.40	366.90
	P17	11503.89	24.04	38.20	464.33	799.22	212.67	724.13	1112.91
	P20	13092.78	13.86	710.86	1288.89	2255.78	621.89	6245.35	8675.24
	P5	1299.00	1.23	0.19	6.33	21.00	4.56	0.15	0.25
	P7	2383.00	4.77	2.27	46.44	93.56	21.78	3.37	4.40
	P10	5757.67	6.70	15.19	152.44	245.44	43.11	23.69	32.75
35	P12	8487.67	6.18	30.55	327.44	488.33	59.78	68.04	98.10
	P15	12175.11	8.64	149.32	881.11	1464.56	326.56	1176.48	1518.99
	P17	13584.22	10.51	242.40	1477.67	2469.44	586.78	2300.48	3278.06
	P20	$\leq$16394.44	11.26	1025.31	$\geq$2516.78	$\geq$4152.33	$\geq$1034.00	$\geq$9896.07	$\geq$14092.73

Table 5: RE-CMCLPC: comparison between smart brute force and our formulation for instance with $B^l = B^f = 15$

Grid Size	Smart B.F.	Our Formulation
5×5	**0.13**	4.52
7×7	**1.16**	18.53
10×10	629.5	**260.28**
12×12	10716.7	**168.54**
15×15	-	**1197.99**

optimal solution for almost all instances in reasonable computational time. The exception was the large instance with budgets $B^l = B^f = 35$, grid 20×20 and $\delta_i = \sqrt{5}$ where the optimal solution was not found after 21600 seconds of execution time.

6.2. Results for the RE-CMCLP

For the RE-CMCLPC we did not test instances with grid 20×20 or instances with budgets $B^l = B^f = 35$. First, we compare our formulation with a brute-force algorithm, testing all possible solutions, in order to compare with the model and prove its accuracy. The algorithm constructs all possible combinations for leader, follower and virtual leader strategies. Some techniques to speed up the process were used, such as quickly discarding the budgets solutions and follower strategies which produce greater regret than the minimum obtained so far. We tested the algorithm only for $B^l = B^f = 15$, given that for $B^l = B^f = 25$ there would be a combinatorial explosion, due to the possibility to place four or even five facilities. Table 5 displays the comparison, where each cell indicates the average time in seconds for all coverage radii tested (1, $\sqrt{2}$, 2, $\sqrt{5}$, $\sqrt{8}$, 3, $\sqrt{10}$, $\sqrt{13}$ and 4). We marked in bold the smallest average time for each type of grid. In addition, we stopped the brute force algorithm after 21600 seconds of execution for instances with grid 15×15 because the algorithm did not reached the optimal solution. The results presented in Table 5 indicate that for instances with grids greater than or equal to 10×10 our model surpasses brute-force performance.

Table 6 shows for each type of instance and each budget, average results

for RE-CMCLPC considering all covering radii. This table has two columns different from Table 4: the columns $Virt \sum w_j$ and $Real \sum w_j$ indicating, respectively, the total demand served by the virtual leader and the leader for the best solutions found. Although the latter values could be different if the instance had more than one optimal solution, this did not happen in our experiments. The solution for the instance with $B^l = B^f = 25$, grid 10×10 and radius $\sqrt{10}$ are just feasible because the algorithm could not find the optimal solution after 21600 seconds of execution. The final gap for that instance was 32.81%. Besides, our reported means do not consider instances with coverage radii greater than or equal to $\sqrt{10}$ for types P12, P15 or P17 with $B^l = B^f = 25$ because no good feasible solution could be found after 21600 seconds of execution.

In Table 6, we observe that the number of separations problems solved as well as the number of generated cuts are small. Nevertheless, even with few separations, the running time for some instances was really high. When we compare the separation time with the total time, we conclude that almost all of the time is spent solving the MILP separations problems. This is due to the complex model of the separations, its large number of constraints and the fact that the objective function is a difference between two terms. We also observe that the number of cuts is close to the number of separations problems solved, indicating that for almost all separation problems solved, a cut is added to the main problem. The average difference between the separation number and the cut number is three, indicating a feasible solution was obtained in those cases. This is due to the fact that the separation problem is only solved for integer relaxed solutions.

6.3. Results for the ST-CMCLP

Comparison with the heuristic from Seyhan (2012). As mentioned before, Seyhan (2012) proposed a heuristic for the ST-CMCLP. The authors proposed a MILP formulation for the general case of the problem. That follower response is based on a greedy heuristic. The authors proposed two additional equivalent formulations to try to improve the relaxation. We implemented the three heuristic variants in order to compare in a fair way our exact method with the heuristic proposed by Seyhan (2012).

The instances used for the comparison are generated similarly to Seyhan (2012). The instances rely on two data sets from Daskin (2011). Both sets are based on the US 1990 census. The first set includes the geographical coordinates and populations of 88 cities. These are the 50 largest US cities

Table 6: RE-CMCLPC: average results considering all covering radii

Instance		Objective Function			Root node		Branch and Bound Tree				
Type	δ_j	Obj	Virt. $\sum w_j$	Real $\sum w_j$	Gap(%)	Time(s)	#Nodes	#Sep	#Cuts	Sep Time(s)	Total Time(s)
15	P5	310.67	2504.11	2193.44	0.00	4.52	0.00	6.00	4.11	4.41	4.52
	P7	795.33	3155.44	2360.11	39.86	17.34	18.56	8.33	6.11	18.03	18.53
	P10	1266.33	6443.11	5176.78	35.83	150.64	26.44	11.56	7.44	256.50	260.28
	P12	1803.56	9278.22	7474.67	42.58	133.87	49.78	10.56	7.89	159.95	168.54
	P15	2565.33	10860.78	8295.44	74.01	197.58	200.44	24.78	19.22	1128.92	1197.99
	P17	2787.11	11290.44	8503.33	81.19	397.50	418.89	34.11	29.11	3348.96	3566.79
25	P5	550.78	2626.89	2076.11	35.28	5.10	70.11	16.44	12.67	7.60	8.01
	P7	871.89	3747.56	2875.67	40.93	46.31	93.67	14.11	11.11	56.45	57.70
	P10	1994.33	7907.00	5912.67	44.21	4015.81	≥584.78	≥30.78	≥25.00	≥8104.70	≥8126.38
	P12	1681.17	9292.50	7611.33	28.27	3526.94	124.33	12.83	9.67	3533.47	3544.50
	P15	2758.83	11606.17	8847.33	43.73	1116.51	1200.67	42.17	36.83	6295.34	6525.54
	P17	3221.33	12097.67	8876.33	71.48	333.81	3855.33	62.33	56.33	4698.64	5843.58

Table 7: ST-CMCLPC: comparison between our method and the heuristic 1 proposed in Seyhan (2012)

Instance Budgets		Heuristic 1 proposed in Seyhan (2012)						Our exact method		
		#opt	Gap(%)		Time (s)			Time (s)		
B^l	B^f		Avg.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.
8	4	39	0.42	3.20	6.42	0.84	41.10	41.78	8.48	141.75
8	5	35	0.55	5.30	20.12	2.11	121.72	69.77	4.00	335.25
8	6	34	0.68	7.31	54.39	4.50	350.42	62.45	2.75	253.14
8	7	32	0.74	6.28	114.84	7.16	475.69	57.22	2.41	339.97
8	8	33	0.68	4.63	219.54	8.47	1495.13	39.36	2.06	278.28

and the 48 continental state capitals, avoiding to count twice any city. The second set includes the geographical coordinates of the 150 largest US cities. The customers are represented by the 88 cities of the first data set where theirs populations represent the customer demands. We considered $|L| = 40$ and $|F| = 20$. Those potential facilities are randomly selected among the 150 cities from the second data set in a way that $L \cap F = \emptyset$. For all instances generated, all fixed costs to open a facility are unitary. We created five group of instances by varying the values of budgets B^l and B^f. For each group, we generated 50 instances. Table 7 shows the comparison. In the table, we present the results associated only to the heuristic 1 because that heuristic presented the best results for the instances tested. The column #opt shows the number of optimal solutions found by the heuristic among the 50 instances generated. In addition, the table presents statistics of the percentage gap for the heuristic and statistics of the CPU time in seconds for both methods.

Observing Table 7, we note that the heuristic found a large number of optimal solutions for all types of instances tested. Moreover, the average percentage gap was less than 1%. However, our method was faster for instances with higher values of the follower budget B^f. In the heuristic, the larger the B^f, the greater is the number of constraints. In our formulation, increasing this parameter makes harder the separation problem (61) – (64). Although the increase of B^f affects both methods, our method was less sensitive to that parameter.

Instances from Plastria and Vanhaverbeke (2008). For the ST-CMCLPC, Table 8 presents for each type of instance and each budget, average results for ST-CMCLPC considering all covering radii. These tables have tree columns that are different from Table 4: the columns *Follower*, *#Sols Calc*, *#Sols Ins* and *#Time Heur(s)* indicating, respectively, the total demand served by the follower, the number of calculated solutions, the number of solutions included and the total time spent by the heuristic that transforms fractional solutions into feasible integer ones described in section 5. For the budget $B^l = B^f = 15$, we set $\epsilon = 0.05$ while for the budgets $B^l = B^f = 25$ and $B^l = B^f = 35$, we set $\epsilon = 0.03$. We stopped the method when it reached 21600s of execution. The time limit was reached for instances of the type P15 with $B^l = B^f = 35$ and coverage radii greater than or equal $\sqrt{8}$. For that reason, we did not test any instance of type P20 and $B^l = B^f = 25$ and any instance of types P17 or P20 with $B^l = B^f = 35$.

Observing Table 8, we reach the following conclusions. First, we observe that the method was able to solve almost all instances in reasonable computational time. The hardest instances were in the type P15 with $B^l = B^f = 35$, where we were not able to obtain the optimal solution in 21600s of execution for five instances. Another point is that the average root gap was not low in some instances. This indicates that the total computational time could be reduced by improving the strength of the formulation. Another important observation is the low average cumulative time spent by solving the separation problems, which was small compared to the average total time for most instances. One last observation concerns the use of the heuristic. Despite the low number of inserted solutions, those were essential to the speed up our method. Several instances were not solved within 3 hours of execution before applying such strategies.

7. Conclusions

This paper presents for the first time ILP formulations for three variants the competitive location problems that are valid for general sets of facilities for the leader and the follower. As the formulations typically contain exponentially many constraints, we generated them on the fly in the course of branch-and-cut algorithms. To show the robustness of the formulations, we compared each formulation proposed here with the one from the literature for particular cases of the problems. The results showed that our algorithms are competitive. Besides we presented results for instances in the general

Table 8: ST-CMCLPC: results for $B^l = B^f = 15$ and small grids

| Instance | | Covered demand | | Root node | | B&B tree | | | | | | | | | | | |
Type	δ_j	Leader	Follower	Gap(%)	Time(s)	#nodes	#Sep Frac	#Sep Total	#Cuts Frac	#Cuts Total	#Sols Calc	#Sols Ins	Time Heur	Time Sep(s)	Time Total(s)
15	P5	1876.00	1561.44	1.21	0.85	1.33	6.22	7.11	4.11	5.00	3.00	1.89	0.11	0.29	0.86
	P7	2768.22	3007.78	6.21	3.63	19.11	10.89	12.44	3.44	4.33	8.33	2.00	1.23	2.02	4.97
	P10	5744.11	6344.44	8.66	24.85	66.00	25.44	27.00	5.00	6.00	21.44	1.44	16.52	26.29	53.43
	P12	8470.56	6789.44	7.32	40.76	184.11	35.11	36.78	6.11	7.00	29.89	1.67	41.68	74.29	151.19
	P15	10348.33	9297.33	5.43	189.60	260.67	36.33	37.89	5.67	6.33	31.33	1.22	130.75	206.06	579.17
	P17	11483.11	10245.67	2.47	323.36	63.11	3.33	6.33	2.00	4.11	3.44	0.67	10.57	26.65	505.73
	P20	12524.78	11911.67	2.36	1016.07	175.22	2.44	7.33	2.00	5.89	4.33	0.78	7.68	43.22	2147.19
25	P5	1440.56	2101.44	6.60	0.76	40.33	13.33	16.11	4.56	6.78	11.00	2.33	0.36	0.72	1.39
	P7	2350.11	3842.89	9.41	6.58	153.11	53.11	56.00	4.56	5.56	49.56	2.33	8.63	12.55	19.44
	P10	5828.44	7457.00	13.06	34.99	2122.67	123.44	127.78	20.33	22.67	105.44	3.89	78.37	136.26	374.61
	P12	9286.00	9063.89	8.91	79.58	2477.22	130.78	138.00	29.56	32.44	104.11	3.11	152.24	265.15	1422.12
	P15	13385.89	12061.56	7.94	231.22	3814.22	57.67	64.44	19.22	24.67	43.89	2.44	217.06	383.94	4146.96
	P17	15523.67	14305.56	5.14	436.71	2180.00	35.89	40.00	12.78	16.00	26.33	2.22	200.03	427.01	4402.13
35	P5	1300.78	2241.22	0.65	0.33	1.78	1.89	4.89	0.89	2.56	2.67	1.56	0.03	0.11	0.35
	P7	2490.44	3917.00	5.34	6.33	220.67	48.89	51.33	10.56	11.78	39.56	3.89	5.72	10.21	17.29
	P10	6210.33	7797.56	9.04	27.25	3057.78	76.33	84.00	27.56	34.11	55.33	4.11	29.48	70.96	333.30
	P12	9369.22	10162.78	7.35	84.03	6741.00	131.78	140.89	46.11	51.89	91.44	4.78	129.07	240.82	2425.77
	P15	14178.00	14341.78	10.01	380.85	9175.22	96.00	103.33	53.00	59.22	49.22	3.67	216.36	468.33	12717.50

case with several sizes, where the formulations could optimally solve large instances in a reasonable amount of time. For future papers, we intend to improve the algorithms performance through two ways: proposing meta-heuristics and finding cuts for both the original and the separation problems. In addition, it could be very interesting to see how our approaches generalize to more general bi-level optimization problems, more particularly for the regret and the Stackelberg variants.

References

Bard, J. F., Moore, J. T., 1992. An algorithm for the discrete bilevel programming problem. Naval Research Logistics (NRL) 39 (3), 419–435.

Church, R., Velle, C. R., 1974. The maximal covering location problem. Papers in regional science 32 (1), 101–118.

Daskin, M. S., 2011. Network and discrete location: models, algorithms, and applications. John Wiley & Sons.

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

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

Eiselt, H., Laporte, G., Thisse, J., 1993. Competitive location models: A framework and bibliography. Transportation Science 27 (1), 44–54.

Fischetti, M., Ljubic, I., Monaci, M., Sinnl, M., 2016. Intersection cuts for bilevel optimization. In: IPCO 2016, Liège, Belgium, June 1-3, 2016, Proceedings. pp. 77–88.

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

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

Israeli, E., Wood, R. K., 2002. Shortest-path network interdiction. Networks 40 (2), 97–111.

Kress, D., Pesch, E., 2012. Sequential competitive location on networks. European Journal of Operational Research 217 (3), 483–499.

Moore, J. T., Bard, J. F., 1990. The mixed integer linear bilevel programming problem. Operations research 38 (5), 911–921.

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

O'Hanley, J. R., Church, R. L., 2011. Designing robust coverage networks to hedge against worst-case facility losses. European Journal of Operational Research 209 (1), 23–36.

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

Roboredo, M. C., Pessoa, A. A., 2013. A branch-and-cut algorithm for the discrete (r| p)-centroid problem. European Journal of Operational Research 224, 101–109.

Serra, D., Ratick, S., ReVelle, C., 1996. The maximum capture problem with uncertainty. Environment and Planning B 23, 49–59.

Serra, D., ReVelle, C., 1994. Market capture by two competitors: The preemptive location problem. Journal of regional Science 34 (4), 549–561.

Seyhan, T. H., 2012. Network design under competitionTheses and Dissertations. Paper 1266.

Taşkın, Z. C., Smith, J. C., Ahmed, S., Schaefer, A. J., 2009. Cutting plane algorithms for solving a stochastic edge-partition problem. Discrete Optimization 6 (4), 420–435.