Expert Systems With Applications 82 (2017) 67–76

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

GRASP and hybrid GRASP-Tabu heuristics to solve a maximal covering location problem with customer preference ordering

Juan A. Díaz[a], Dolores E. Luna[a,b,*], José-Fernando Camacho-Vallejo[c], Martha-Selene Casas-Ramírez[c]

[a] Depto. de Actuaría, Física y Matemáticas, Universidad de las Américas Puebla, México
[b] Departamento de Ingeniería Industrial y Mecánica, Universidad de las Américas Puebla, Ex Hacienda Santa Catarina Mártir. CP. 72810, San Andrés Cholula, Puebla, México
[c] Facultad de Ciencias Físico Matemáticas, Universidad Autónoma de Nuevo León, México

ARTICLE INFO

Article history:
Received 22 June 2016
Revised 31 March 2017
Accepted 1 April 2017
Available online 2 April 2017

Keywords:
Maximal covering
Metaheuristics
GRASP
Tabu search
Bilevel programming

ABSTRACT

In this study, a maximal covering location problem is investigated. In this problem, we want to maximize the demand of a set of customers covered by a set of *p* facilities located among a set of potential sites. It is assumed that a set of facilities that belong to other firms exists and that customers freely choose allocation to the facilities within a coverage radius. The problem can be formulated as a bilevel mathematical programming problem, in which the leader locates facilities in order to maximize the demand covered and the follower allocates customers to the most preferred facility among those selected by the leader and facilities from other firms. We propose a greedy randomized adaptive search procedure (GRASP) heuristic and a hybrid GRASP-Tabu heuristic to find near optimal solutions. Results of the heuristic approaches are compared to solutions obtained with a single-level reformulation of the problem. Computational experiments demonstrate that the proposed algorithms can find very good quality solutions with a small computational burden. The most important feature of the proposed heuristics is that, despite their simplicity, optimal or near-optimal solutions can be determined very efficiently.

1. Introduction

In this study, we consider a variant of the maximal covering location problem. We study a situation in which a firm wants to locate a number of *p* facilities to maximize the captured demand in a market where a number of firms already operate. We identify two decision levels in this problem: the location of facilities and the allocation of customers. Typically, in most location problems, both decisions are assumed to be made by a single decision maker. However, there are situations wherein customers are free to choose a service provider, particularly in cases where customers travel to service providers' facilities. In most location-allocation problems in the literature, customers are assigned to the nearest facility. However, customer behavior is influenced by many factors, such as age, income, education, etc.; therefore, customers do not always go to the nearest facility. We consider that there is a threshold distance that represents the maximum distance a customer is willing to travel to obtain a service. Therefore, a customer will patronize their most preferred facility among those within the threshold distance. We model customer preference ordering as in Belotti, Labbé, Maffioli, and Ndiaye (2007); Cánovas, García, Labbé, and Marín (2007); Hanjoul and Thill (1987); Hansen, Kochetov, and Mladenovi (2004). For each customer, a list indicating the order of preference for each potential facility within the threshold distance is used. Ognjanović, Gaševi, and Bagheri (2013) suggest that an ordered list of customer's facility preferences could be considered for many real world applications in diverse fields/areas. They propose ordering preferences in a lexicographic manner in the allocation variables, which is very similar to the ordered list of preferences used in our research. We also consider that the competitive model is static, as defined in Plastria (2001), where it is assumed that the competition will not react in the short term due to economic constraints or the time required to react.

The maximal covering location problem (introduced in Church & ReVelle, 1974) locates a fixed number of *p* facilities from a set of potential sites to maximize customer demand covered within a coverage radius. In this paper, we study an extension of the max-

* Corresponding author at: Departamento de Ingeniería Industrial y Mecánica, Universidad de las Américas Puebla, Ex Hacienda Santa Catarina Mártir. CP. 72810, San Andrés Cholula, Puebla, México.

E-mail addresses: juana.diaz@udlap.mx (J.A. Díaz), dolorese.luna@udlap.mx (D.E. Luna), jose.camachovl@uanl.edu.mx (J.-F. Camacho-Vallejo), martha.casasrm@uanl.edu.mx (M.-S. Casas-Ramírez).

http://dx.doi.org/10.1016/j.eswa.2017.04.002
0957-4174/

imal covering location problem, where it is assumed that a set of facilities that belong to other firms exists and that customers can choose the facility that will provide the service within a coverage radius. This problem can be formulated as a bilevel mathematical programming problem where the leader determines facility locations from a set of potential sites and the follower then determines the allocation of customers to their most preferred sites. According to an extensive literature review, only Lee and Lee (2012) consider customer preferences in relation to coverage problems. In that paper, the goal is to maximize the weighted demand covered with respect to a customer preference index. The authors also consider constraints with respect to upper and lower bounds on the number of customers allocated to each facility.

Many location problem studies have considered customer preference. For example, Hansen et al. (2004) considered user preferences in relation to the uncapacitated facility location problem. In this problem, the leader's objective function is to minimize location and distribution costs and the follower's objective function optimizes customer preferences. In that study, the authors proposed a new reformulation and showed that it provides better linear programming (LP) relaxations compared to three reformulations from the literature. Based on experiments using random data instances, Hansen et al. (2004) concluded that the proposed reformulation can find an optimal solution faster than the compared approaches. Another single-level reformulation of the same problem was proposed by Vasil'ev, Klimentova, and Kochetov (2009). In that reformulation, clique inequalities were used to obtain lower bounds. Later, Vasilyev and Klimentova (2010) employed linear relaxation of the bilevel problem to obtain lower bounds and applied a simulated annealing heuristic to obtain feasible solutions. A branch and cut algorithm that combines both bounding schemes was used to obtain near optimal solutions. Metaheuristic methods have also been used to obtain solutions for this problem. A Stackelberg evolutionary algorithm was proposed in Camacho-Vallejo, Cordero-Franco, and González-Ramírez (2014). In that study, good quality feasible solutions for up to 500 facilities and 1000 customers were obtained efficiently. In addition, Marić, Stanimirović, and Milenković (2012) proposed different metaheuristics to obtain feasible solutions for the problem, i.e., a particle swarm optimization method, a simulated annealing method, and an adapted neighborhood search method. In these methods, the lower level problem can be obtained optimally by rearranging the customer preference matrix.

Bilevel competitive facility location problems have also been considered in the literature. Kress and Pesch (2012) surveyed sequential competitive location problems. They divided the problems into two categories (see Hakimi, 1983): $(r|X_p)$-medianoid and $(r|p)$-centroid problems. The problems were formulated using a leader–follower approach. In the $(r|X_p)$-medianoid problem, given that the leader has p facilities, the follower optimally locates r facilities from a set of potential sites X_p. In the $(r|p)$-centroid problem, the leader optimally locates p facilities knowing that the follower will react by locating r facilities.

In this study, we propose a greedy randomized adaptive search procedure (GRASP) heuristic and a hybrid GRASP-Tabu heuristic. The hybrid GRASP-Tabu heuristic combines GRASP and tabu search to efficiently find lower bounds for large-scale instances for the maximal covering location problem with customer preference ordering. To evaluate the quality of the obtained bounds, we reformulate the problem as a single-level integer programming problem using valid inequalities, as in Cánovas et al. (2007), to ensure that each customer is allocated to the most preferred facility among those that are opened. According to computational results, despite their simplicity, the proposed algorithms provide optimal or near optimal solutions in a small computation time.

The remainder of this paper is organized as follows. In Section 2, a bilevel formulation of the problem is presented and an equivalent single-level reformulation is proposed. In Section 3, the GRASP and the hybrid GRASP-Tabu heuristics are described. Computational experiments are reported in Section 4, and conclusions are presented in Section 5.

2. Problem statement

In this section, we describe the Bilevel Maximal Covering Location Problem (*BLMCLP*), a bilevel mathematical formulation of the problem, and a single-level reformulation that replaces the follower's decision problem with a set of valid inequalities. That replacement guarantees that customer demands are allocated to facilities located at the most preferred sites.

The *BLMCLP* considers the following situation: a firm is intending to enter the market by opening p facilities, which are to be located within a predefined set of potential sites, to capture the demand of a set of customers who are free to patronize any facility among those that already exist or those to be opened. It is assumed that the firm knows customer preferences. Then, we can consider that the leader is the entering competing firm. In this situation, the leader determines the location of exactly p facilities to maximize the captured demand and the follower is a set of customers that will choose which facility will provide service according to their preferences and the coverage radius. In addition, we consider that the existing facilities in the market cannot be closed.

The situation described above could be modelled with bilevel programming since the firm (leader) determines facility locations and the customers (follower) patronize their most preferred facility among open facilities. Thus, due to the decision-making hierarchy and the lack of cooperation between decision makers, the problem can be formulated as a bilevel programming problem.

2.1. Mathematical bilevel formulation

Let I_1 be the set of potential sites to locate the facilities and I_2 be the set of existing facility sites, where $I = I_1 \cup I_2$. Let J_2 be the set of customers covered by an existing facility and J_1 be the set of customers not covered by existing facilities (i.e., there is not an existing facility within its coverage radius), where $J = J_1 \cup J_2$. For each $j \in J$, we define $I(j)$ as the index set of facilities that cover the demand of customer j (i.e., the set of potential locations not exceeding a maximal distance from the customer location) and, for each $i \in I$, we define $J(i)$ as the index set of customers whose demand can be covered by a facility located at site i. Let D_j be the demand associated with customer $j \in J$, p be the number of facilities that the leader wishes to locate, and g_{ij} be the preference of customer $j \in J$ toward the facility located at $i \in I$.

Consider the following decision variables:

$$y_i = \begin{cases} 1, & \text{if a facility is located in site } i \\ 0, & \text{otherwise} \end{cases}, i \in I$$

and

$$x_{ij} = \begin{cases} 1, & \text{if customer } j \text{ is allocated to} \\ & \text{facility located in site } i \\ 0, & \text{otherwise} \end{cases}, j \in J, i \in I(j)$$

The *BLMCLP* can be formulated as a bilevel program as follows.

$$(BLMCLP) \quad \max_{y,x} \sum_{i \in I_1} \sum_{j \in J(i)} D_j x_{ij} \tag{1}$$

$$\text{subject to:} \quad y_i = 1 \quad \forall i \in I_2 \tag{2}$$

$$\sum_{i \in I_1} y_i = p \tag{3}$$

$$y_i \in \{0, 1\} \quad \forall i \in I \tag{4}$$

$$\text{where} \quad x \quad \text{solves}$$

$$\max_x \quad \sum_{i \in I} \sum_{j \in J(i)} g_{ij} x_{ij} \tag{5}$$

$$\text{subject to:} \quad \sum_{i \in I(j)} x_{ij} = 1 \quad \forall j \in J_2 \tag{6}$$

$$x_{ij} \le y_i \quad \forall i \in I, j \in J(i) \tag{7}$$

$$\sum_{i \in I(j)} x_{ij} \le 1 \quad \forall j \in J_1 \tag{8}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J(i) \tag{9}$$

In this formulation, the leader's objective function (1) maximizes the demand covered by the new p facilities. Constraints (2) and (3) ensure that the existing facilities remain open and that exactly p new facilities will be located, respectively. The values of the decision variables x_{ij}, $i \in I$, $j \in J(i)$, represent the optimal solution of the follower's problem defined by (5)–(9). The follower's objective function (5) maximizes the sum of customer preferences. Constraints (6) guarantee that customer demand that is covered by an existing facility will remain covered by the same facility or a new facility. Constraints (7) ensure that customer demand can only be covered by an open facility located within its coverage set. Finally, constraints (8) ensure that uncovered customer demands might remain uncovered if no facility is located within their coverage radius.

A well-defined bilevel problem relies on the existence and uniqueness of an optimal solution for the lower level problem (Vasil'ev et al., 2009). In other words, if alternative optimal solutions exist for the follower's problem for a given leader's decision, the bilevel problem is ill-posed. In the problem considered here, the follower's problem will have a unique optimal solution if for each customer $j \in J$, their preferences are positive consecutive numbers from 1 to $|I(j)|$.

2.2. Single-level reformulation

As mentioned in Bard (1998), Colson, Marcotte, and Savard (2007), and Kalashnikov, Dempe, Pérez-Valdéz, Kalashnikova, and Camacho-Vallejo (2004), solving a bilevel problem may not be an easy task; even a linear bilevel problem is NP-hard (see Jeroslow, 1985 and Hansen, Jaumard, & Savard, 1992). Furthermore, since there is no general methodology and thus no mathematical programming software to solve any bilevel problem, specific methods to obtain optimal or near optimal solutions need to be explored. The *BLMCLP* previously described can be reduced to a single-level mixed integer programming problem. In classical approaches, the reformulations are performed using the primal-dual relationships for the follower's problem. To ensure optimality for the follower's problem, we can force the equality of the objective functions of both primal and dual problems or include complementary slackness constraints. In this study, we considered these two reformulations and a direct single-level reformulation that ensures the optimality of the follower's problem, which forces customer demand to be allocated to the most preferred facility. After performing preliminary computational tests with the three reformulations, it was observed that the linear relaxation bounds are much better for the proposed reformulation. We describe the single-level maximal covering location problem (*SLMCLP*) reformulation in the following.

Since the follower's problem ensures that the demand of customers that are within the coverage radius of one or more facilities is allocated to the most preferred facility, the bilevel mathematical program can be formulated as a single-level program. As in Cánovas et al. (2007), we consider a single-level reformulation of the problem by replacing the follower's decision problem with a set of valid inequalities that ensure that whenever a customer is within the coverage radius of more than one facility, customer demand will always be allocated to the most preferred facility.

For each $j \in J$, let $i_1, i_2, \ldots, i_{|I(j)|}$ be the elements in $I(j)$ such that $g_{i_1,j} > g_{i_2,j} > \ldots > g_{i_{|I(j)|},j}$. Then, to ensure that the demand of each customer is allocated to the most preferred facility, we include the following valid inequalities.

$$\sum_{s=k+1}^{|I(j)|} x_{i_s,j} + y_{i_k} \le 1 \qquad \forall j \in J, k \in 1, \ldots, |I(j)| - 1 \tag{10}$$

These inequalities ensure that if there is no open facility in the kth preferred location for customer j, then the customer must be allocated to a facility whose location is less preferred than the kth location or their demand is not allocated to any open facility. Then, the problem can be modelled with the following single-level mathematical programming model.

$$(SLMCLP) \quad \max_{y,x} \quad \textstyle\sum_{i \in I_1} \sum_{j \in J(i)} D_j x_{ij} \tag{1}$$

$$\text{subject to:} \quad y_i = 1 \quad \forall i \in I_2 \tag{2}$$

$$\textstyle\sum_{i \in I_1} y_i = p \tag{3}$$

$$y_i \in \{0, 1\} \quad \forall i \in I \tag{4}$$

$$\textstyle\sum_{i \in I(j)} x_{ij} = 1 \quad \forall j \in J_2 \tag{6}$$

$$x_{ij} \le y_i \quad \forall i \in I, j \in J(i) \tag{7}$$

$$\textstyle\sum_{i \in I(j)} x_{ij} \le 1 \quad \forall j \in J_1 \tag{8}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J(i) \tag{9}$$

$$\textstyle\sum_{s=k+1}^{|I(j)|} x_{i_s,j} + y_{i_k} \le 1 \quad \forall j \in J, k \in 1, \ldots, |I(j)| - 1 \tag{10}$$

In the above formulation, we observe that constraints $x_{ij} \in \{0, 1\}$, $i \in I$, $j \in J(i)$ can be replaced by $x_{ij} \ge 0$, $i \in I$, $j \in J(i)$. Given any feasible solution, where $I^* = \{i \in I : y_i = 1\}$ denotes the set of open facilities, for a given customer $j \in J$, let $i_1, i_2, \ldots, i_{|I(j)|}$ such that $g_{i_1 j} > g_{i_2 j} > \ldots > g_{i_{|I(j)|} j}$. Then, if $x_{i_k j} > 0$ by constraints (7), this implies that $y_{i_k} = 1$. In addition, by constraints (10) and (7), $y_{i_r} = 0$ and $x_{i_r j} = 0$ for all $r = 1, \ldots, k-1$. Otherwise, $x_{i_k j}$ cannot be greater than zero because if $y_{i_r} = 1$, for some $r = 1, \ldots, k-1, x_{i_k j}$ should be zero to satisfy constraints (10) associated with facility location i_r and customer j. Finally, $x_{i_r j} = 0$ for all $r = k+1, \ldots, |I(j)|$ given that $y_{i_k} = 1$. Therefore, by constraints (6) or constraints (8) and assuming that $D_j > 0, x_{i_k j} = 1$.

Here, we prove the equivalence between *BLMCLP* and *SLMCLP*.

Proposition 1. *If $(\bar{x}, \bar{y})$ is a feasible solution for BLMCLP, then valid inequalities (10) are satisfied and therefore $(\bar{x}, \bar{y})$ is also feasible for SLMCLP.*

Proof. Let $\bar{x}$ be the optimal solution for the lower level problem for a given feasible solution of the leader's solution $\bar{y}$.

Therefore, in any feasible solution of the bilevel problem, for each customer $j \in J$ and a given $k \in [1, |I(j)|]$, we have the following cases.

1. $\bar{y}_{i_k} = 0$. We observe that in any feasible solution of *BLMCLP*, $\forall j \in J$, $\sum_{i \in I(j)} x_{ij}$ is at most 1 since if $j \in J_2$, then $\sum_{i \in I(j)} \bar{x}_{ij} = 1$ and if $j \in J_1$, then $\sum_{i \in I(j)} \bar{x}_{ij} \leq 1$. Therefore, $\sum_{s=k+1}^{|I(j)|} \bar{x}_{i_s,j} \leq 1$ since the set $\{i_{k+1}, \ldots, i_{|I(j)|}\} \subset I(j)$.
2. $\bar{y}_{i_k} = 1$. Therefore, $\bar{x}_{i_s,j} = 0$ for all $s > k$ since constraints (6)–(8) ensure that the demand of each customer is allocated to at most one open facility and if $\bar{x}_{i_s,j} = 1$ for some $s > k$, this will contradict the optimality of the lower level problem. In addition, since it is an uncapacitated problem and $g_{i_k,j} > g_{i_s,j}$ for all $s > k$, the demand of customer j can be allocated to their kth preferred location because there is a facility located at that site (i.e., $\bar{y}_{i_k} = 1$).

Therefore, in both cases, constraints (10) hold. □

Proposition 2. *Let* $(x^\star, y^\star)$ *be an optimal solution of SLMCLP. Then,* $x^\star$ *is the optimal solution of the lower level problem in BLMCLP for the given* $y^\star$*; therefore,* $(x^\star, y^\star)$ *is feasible for BLMCLP.*

Proof. Let $I^\star = \{i \in I : y_i^\star = 1\}$. For each $j \in J$, we have the following mutually exclusive cases.

1. If $j \in J_2$, then the demand of customer j must be allocated to exactly one open facility because their demand is already covered by existing facilities. Therefore, let i_k be the facility to which the demand of customer j is allocated, i.e., $x^\star_{i_k,j} = 1$ for some $k \in [1, |I(j)|]$. Then, $x^\star_{i_s,j} = 0$ for all $s \in I(j) \backslash \{k\}$ because constraints (6) must hold. In addition, $y^\star_{i_k} = 1$ because $x^\star_{i_k,j} \leq y^\star_{i_k}$. Constraints (10) ensure that $y^\star_{i_s} = 0$ for all $s < k$ since if $y^\star_{i_s} = 1$ for some $s < k$, $x^\star_{i_k,j}$ must be equal to 0. Therefore, the demand of customer j is allocated to their most preferred facility among the open facilities, i.e., $i_k = \arg\max_{i \in I(j) \cap I^\star} \{g_{ij}\}$.
2. If $j \in J_1$, then the demand of customer j must be allocated to at most one open facility because constraints (8) must hold. Therefore, in the same way as in the case where $j \in J_1$, if the customer demand is allocated to facility i_k, this means that $x^\star_{i_k,j} = 1$ for some $k \in [1, |I(j)|]$ and $x^\star_{i_s,j} = 0$ for all $s \in I(j) \backslash \{k\}$. In addition, $\bar{y}_{i_k} = 1$ since $x^\star_{i_k,j} \leq y^\star_{i_k}$; thus, constraints (10) ensure that $y_{i_s} = 0$ for all $s < k$ and $i_k = \arg\max_{i \in I(j) \cap I^\star} \{g_{ij}\}$. If the demand of customer j is not allocated, this means that $y^\star_i = 0$ and $x^\star_{ij} = 0, \forall i \in I(j)$ because if $y^\star_k = 1$ for at least one $k \in I(j)$, this will contradict the optimality of the *SLMCLP* because demands are assumed to be positive and therefore the objective value can be increased by D_j by setting $x^\star_{kj} = 1$. □

Proposition 3. *BLMCLP and SLMCLP provide the same optimal solutions.*

Proof. Let $(x^\star, y^\star)$ be an optimal solution for *BLMCLP*. By Proposition 1, we know that this solution is also feasible for *SLMCLP*. Assume that there exists an optimal solution $(\bar{x}, \bar{y})$ for the single-level formulation such that

$$\sum_{i \in I_1} \sum_{j \in J(i)} D_j \bar{x}_{ij} > \sum_{i \in I_1} \sum_{j \in J(i)} D_j x^\star_{ij}.$$

By Proposition 2, we know that this solution is also feasible for the bilevel formulation; therefore, it contradicts the optimality of $(x^\star, y^\star)$ for *BLMCLP*.

Conversely, let $(x^\star, y^\star)$ be an optimal solution for *SLMCLP*. By Proposition 2, we know that this solution is also feasible for *BLMCLP*. Assume that there is another feasible solution $(\bar{x}, \bar{y})$ for *BLMCLP* such that

$$\sum_{i \in I_1} \sum_{j \in J(i)} D_j \bar{x}_{ij} > \sum_{i \in I_1} \sum_{j \in J(i)} D_j x^\star_{ij}.$$

By Proposition 1, $(\bar{x}, \bar{y})$ is also feasible for the *SLMCLP*, thereby contradicting the optimality of $(x^\star, y^\star)$ for the *SLMCLP*. □

From the latter results, it can be concluded that both mathematical formulations are equivalent.

Note that this single-level reformulation can be used to obtain a reference value in order to measure the performance of the proposed algorithms.

3. GRASP and hybrid GRASP-Tabu heuristics

Artificial intelligence (AI) has been applied in many domains, such as robotics, speech recognition, planning and programming for many tasks, logistics planning, pattern recognition, and VLSI design (Russell & Norvig, 2010). When attempting to provide solutions to such problems, AI encounters many obstacles. One obstacle is related to the fact that many of the problems involve combinatorial problems in which finding all possible alternative solutions and investigating them is, in practical terms, impossible. In such situations, it is very useful to have informed search strategies that enable more efficient search processes. Heuristic and metaheuristic methods help manage these complex decision problems by taking advantage of the structure and characteristics of the problems to be solved, thereby providing AI with efficient search methods (Glover, 1989).

The GRASP metaheuristic was first proposed by Feo and Resende (1995). It is an iterative search procedure. In each iteration, a new solution is constructed by a greedy randomized procedure, which is then improved using a local search procedure. In Feo and Resende (1995), the authors give an intuitive justification for GRASP as a repetitive sampling technique. In each iteration of the greedy algorithm, a new element is selected from a *restricted candidate list* and added to the solution. The mean and variance of the sample distribution are a function of the cardinality of the restricted candidate list used in the constructive phase of GRASP. Since sample solutions are selected randomly, by order statistics, one can intuitively expect that the best value found should outperform the mean value.

Tabu search (Glover, 1989; 1990) relies on the use of adaptive memory and responsive exploration. It uses adaptive memory to record historical information of the search process. The term responsive exploration refers to the ability of the method to make strategic choices to achieve effectiveness.

Memory structures used by tabu search operate by referencing four dimensions: recency, frequency, quality, and influence. The simplest forms of tabu search only use recency-based memory to avoid cycling. The memory structures are used to record the attributes of solutions recently visited, which are labeled tabu-active, and solutions that contain tabu-active attributes are forbidden. Therefore, the use of recency-based memory is an aggressive exploration strategy that attempts to go beyond local optimality because it prevents revisiting solutions visited in the recent past. Aspiration criteria can also be used to override tabu restrictions (remove tabu-active attributes). The simplest form of an aspiration criterion is to remove the tabu classification of a solution that is better than the best solution found so far.

Long-term memory is often used by a tabu search procedure for intensification or diversification purposes. Intensification strategies can be used to exploit features historically found to be good, while diversification strategies encourage the search process to explore unvisited regions of the solution space. Commonly, frequency-based memory is used to implement intensification and diversification strategies.

The quality dimension refers to the ability to differentiate the merit of solutions visited during the search, whereas the influence dimension refers to the impact of the choices made during the search process relative to both quality and structure of the visited solutions.

In the following, we describe the two proposed heuristics to find lower bounds for *BLMCLP*: a GRASP heuristic and a hybrid GRASP-Tabu heuristic, which is a combination of GRASP and tabu search.

3.1. GRASP heuristic

The proposed GRASP heuristic iteratively constructs an initial feasible solution that is subsequently improved by a local search that explores one neighborhood. The incumbent solution is updated each time an iteration obtains a better solution. The algorithm terminates when the termination criterion is met. Here, we first describe the greedy randomized procedure used in the proposed GRASP heuristic and then describe the local search procedure.

3.1.1. Greedy randomized procedure

To find initial feasible solutions to *BLMCLP*, a set $S \subset I_1$ of p facilities must be selected such that customer demand covered by the firm is maximized. The greedy randomized procedure of the proposed GRASP heuristic is an iterative procedure that constructs an initial feasible solution. In each iteration, a facility from the set of potential facilities I_1 is selected and covered customers are assigned to their most preferred facility among the existing facilities. A greedy function is used to evaluate the contribution to the leader's objective function of each candidate facility in I_1. For each candidate facility, the greedy function measures the additional demand that can be covered by the facility by considering the follower's reaction. The demand that can be captured by a candidate facility is the demand of customers within its coverage radius and that is not covered by any open facility or a demand that is covered by some facility in I_2 but the customer prefers the candidate facility among all open facilities. Next, we explain the constructive procedure in detail.

Let *Covered* be the set of customers covered by open facilities. Initially, *Covered* is equal to J_2. In addition, A: $J \to S \cup I_2$, where $A(j) := i$ if customer j is allocated to facility i. For each $j \in J$, $A(j) := \arg\max_{i \in (S \cup I_2) \cap I(j)} \{g_{ij}\}$, if $(S \cup I_2) \cap I(j) \neq \emptyset$. Otherwise, $A(j)$:= *null*, which means that customer j is not within the coverage radius of any open facility and therefore $j \notin Covered$. Initially, S is equal to $\emptyset$ and it is updated at the end of each iteration. To select the facility to be open at each iteration, the greedy value w_i is computed for each potential facility $i \in I_1 \backslash S$. As mentioned previously, this greedy value w_i measures the increase in the leader's objective function value to open a facility at location i. In other words, for each customer j covered by i (i.e., $j \in J(i)$), the demand is added to w_i in either of the following two situations: 1) j was not previously covered, $j \notin Covered$, and 2) j was covered by a facility $A(j) \in I_2$ and i is more preferred than $A(j)$. For each $i \in I_1 \backslash S$, we have the following.

$$\beta_{ij} = \begin{cases} D_j & \text{if } j \notin Covered \text{ or } (g_{ij} > g_{A(j),j} \text{ and } A(j) \in I_2) \\ 0 & \text{otherwise} \end{cases}, \forall j \in J(i)$$

therefore

$$w_i = \sum_{j \in J(i)} \beta_{ij} \tag{11}$$

Let $w_{\min} = \min_{i \in I_1 \backslash S} \{w_i\}$ and $w_{\max} = \max_{i \in I_1 \backslash S} \{w_i\}$. In addition, let $T = w_{\min} + \alpha(w_{\max} - w_{\min})$ be a threshold value, where $\alpha \in (0, 1)$ is a parameter that controls the degree of greediness or randomness of the greedy procedure. We randomly select a facility i^* from a restricted candidate list *RCL* that contains candidate facilities whose $w_i \geq T$. Facility i^* is added to the set of open facilities S. At the end of each iteration, the *Covered* set is updated and customers $j \in J(i^*)$ are allocated to i^* if they were not previously covered or if i^* is more preferred than its previous assignment $A(j)$. In addition, $A(j)$ is updated for all $j \in Covered$. In other words, *Covered* is set to $\bigcup_{i \in (S \cup I_2)} J(i)$, which is the set of all customers within the coverage radius of some open facility. For all $j \in Covered$, $A(j)$ is set to $\arg\max_{i \in (S \cup I_2) \cap I(j)} \{g_{ij}\}$, which means that each covered customer is allocated to their most preferred location among all locations in $S \cup I_2$. The constructive phase ends when p facilities are selected. The randomized greedy procedure of the proposed GRASP heuristic is given in Algorithm 3.1.

Algorithm 3.1 Greedy randomized procedure.

function GreedyRandomized(α)
  $SolValue \leftarrow 0$
  $selected \leftarrow 0$
  $S \leftarrow \emptyset$
  $Covered \leftarrow J_2$
  for all $j \in J_2$ **do**
    $A(j) \leftarrow \arg\max_{i \in I_2} \{g_{ij} : j \in J(i)\}$
  end for
  repeat
    for all $i \in I_1 \setminus S$ **do**
      for all $j \in J(i)$ **do**

$$\beta_{ij} \leftarrow \begin{cases} D_j & \text{if } j \notin Covered \text{ or } (g_{ij} > g_{A(j),j} \text{ and } A(j) \in I_2) \\ 0 & \text{otherwise} \end{cases}$$

      end for
      $w_i \leftarrow 0$
      for all $j \in J_1$ **do**
        $w_i \leftarrow w_i + \beta_{ij}$
      end for
    end for
    $w_{\min} \leftarrow \min_{i \in I_1 \setminus S} \{w_i\}$
    $w_{\max} \leftarrow \max_{i \in I_1 \setminus S} \{w_i\}$
    $T \leftarrow w_{\min} + \alpha(w_{\max} - w_{\min})$
    $RCL \leftarrow \{i \in I_1 \setminus S : w_i \geq T\}$
    select i^* randomly from *RCL*
    $S \leftarrow S \cup \{i^*\}$
    $selected \leftarrow selected + 1$
    $Covered \leftarrow \bigcup_{i \in (S \cup I_2)} J(i)$
    $A(j) \leftarrow \arg\max_{i \in (S \cup I_2) \cap I(j)} \{g_{ij}\}$
    $DemandCovered \leftarrow DemandCovered + w_{i^*}$
  until $selected = p$
  return *SolValue*
end function

3.1.2. Local search procedure

Each initial solution generated by the randomized greedy procedure of the GRASP heuristic is improved by a local search procedure. In this manner, the local search procedure in a GRASP heuristic is used as an intensification strategy (Resende & González-Velarde, 2003). In our case, the local search procedure explores the neighborhood wherein one open facility is exchanged with a closed facility, i.e., the exchange opens a closed facility $i_1 \in I_1 \backslash S$ and closes an open facility $i_2 \in S$. A solution to the maximal covering location problem with customer preference ordering is represented by a pair (S, A), where S is the set of open facilities and A: $J \to S \cup I_2$, as defined in the previous section. Therefore, the neighborhood N(S, A) of a given solution (S, A) is expressed as follows.

$$N(S,A) = \Big\{(S', A') : S' = S \setminus \{i_2\} \cup \{i_1\}, i_2 \in S, i_1 \in I_1 \setminus S,$$
$$A'(j) = \arg\max_{i \in (S' \cup I_2) \cap I(j)} \{g_{ij}\}, \forall j \in \cup_{i \in (S' \cup I_2)} J(i),$$
$$A'(j) = null, \forall j \notin \cup_{i \in (S' \cup I_2)} J(i)\Big\}$$

Here, N(S, A) is the set of all solutions that can be obtained by opening a closed facility and closing an open facility.

Let

$$\delta_j^{i_1,i_2} = \begin{cases} -D_j, & \text{if } A(j) = i_2 \text{ and } (j \notin \cup_{i \in S \setminus \{i_2\} \cup \{i_1\}} J(i) \\ & \quad \text{or } \arg\max_{i \in (S \cup I_2) \setminus \{i_2\} \cup \{i_1\}} \{g_{ij}\} \in I_2) \\ D_j, & \text{if } j \in J(i_1) \text{ and } (j \notin \cup_{i \in (S \cup I_2)} J(i)) \\ & \quad \text{or } (A(j) \in I_2 \text{ and } g_{A(j),j} < g_{i_1,j})) \\ 0, & \text{otherwise} \end{cases}$$

for all $j \in J$, denote the leaders' objective function change associated with customer j when we exchange facility i_1 with facility i_2. As can be seen, the objective function value decreases by D_j if the demand of customer j is covered by facility i_2 in solution (S, A), i.e., $A(j) = i_2$ and

- the demand of customer j cannot be covered by the set of the leader's opened facilities $S \backslash \{i_2\} \cup \{i_1\}$ after the exchange is performed, or
- the demand of customer j will be allocated to some facility in I_2 according to preferences (i.e., $\arg\max_{i \in (S \cup I_2) \setminus \{i_2\} \cup \{i_1\}} \{g_{ij}\} \in I_2$).

On the other hand, the objective function value will increase by D_j when the demand of customer j is covered by facility i_1 (i.e., $j \in J(i_1)$) and

- the demand of customer j is not covered in the current solution (i.e., $j \notin \cup_{i \in (S \cup I_2)} J(i)$), or
- the demand of customer j is allocated to some facility in I_2; however, according to customer preferences, it will be reallocated to facility i_1 because $g_{A(j),j} < g_{i_1,j}$.

Then, the leader's objective function change associated with the exchange of facility i_1 with facility i_2 is expressed as follows.

$$\Delta_{i_1,i_2} = \sum_{j \in J} \delta_j^{i_i,i_2}$$

Here, Δ_{i_1,i_2} measures the impact to the leader's objective function value by computing the customer's demand change when facility $i_1 \in I_1 \backslash S$ is opened and the impact to the customer's demand change when facility $i_2 \in S$ is closed by considering the follower's reaction with respect to customer preferences.

The local search of the proposed GRASP heuristic uses the best improvement strategy (i.e., the best solution in N(S, A) is selected) and terminates when the objective function value cannot be improved further (all solutions in N(S, A) are worse than the current solution). The local search procedure of the proposed GRASP heuristic is given in Algorithm 3.2.

The proposed GRASP heuristic is given in Algorithm 3.3. The termination criterion is a predetermined number of iterations without improvement.

3.2. Hybrid GRASP-Tabu heuristic

Hybrid algorithms have been widely used to solve difficult problems (Blum, 2010; Lozano & García-Martínez, 2010; Talbi, 2002). These combinations enhance the advantages of using a single methodology to obtain better solutions. The combination of GRASP with Tabu search was first studied in Laguna and González-Velarde (1991). Hybrid methods using GRASP with Tabu search have been used to find feasible bounds for many problems (Abdinnour-Helm & Hadley, 2000; Colomé & Serra, 2001; Delmaire, Díaz, Fernández, & Ortega, 1999; Díaz, Luna, & Zetina, 2013; Duarte & Martí, 2007; Laguna & González-Velarde, 1991; Lim & Wang, 2004). In this study, we propose a procedure that combines GRASP and Tabu search. The Tabu search procedure is used instead of the local search in the algorithm's improvement phase.

Tabu search is commonly used to solve difficult problems. It was first proposed in Glover (1986) and it is described in detail in Glover (1989). In this methodology, short-term and long-term memory are used to escape local optima in order to better explore the solution space. Three elements must be defined to implement tabu search: tabu attributes, tabu-tenure (a parameter that indicates the number of iterations in which an attribute will be active), and the aspiration criterion (criterion to override tabu-active attributes). Tabu-active attributes are those that are forbidden in a solution. The short-term memory of a tabu search contains a list of selected attributes that occur in recently visited solutions to prevent revisiting these solutions.

In our implementation, the tabu attributes are defined in a simple manner. Once a facility $i \in S$ is closed, it must remain closed for a number of iterations. To implement this, a fixed size list with a first in-first out discipline is maintained. Therefore, if facility $i^\star \in S$ is closed at iteration t, the index of facility $i^\star$ will be appended to the tabu list. If the fixed size of the tabu list is r, opening facility $i^\star$ will be forbidden in iterations $t+1, t+2, \ldots, t+r$. We also use the standard aspiration criterion every time a solution with a tabu-active attribute is better than the incumbent solution, i.e., the tabu restrictions are ignored. Therefore, at any iteration of the procedure, the candidate neighbor solutions will be those that do not contain tabu-active attributes or satisfy the aspiration criterion (line 14 in Algorithm 3.4). The proposed hybrid GRASP-Tabu heuristic is given in Algorithm 3.5. The termination criterion is a predetermined number of iterations without improvement.

4. Computational experimentation

To evaluate the performance of the proposed heuristic methods, we used 60 randomly generated instances. The instances were generated by the procedure described in Resende (1998). The procedure used in this study can be summarized as follows: (1) t points are generated in a unit square and the matrix of Euclidean distances is computed, (2) m of the t points are selected randomly as potential facility locations and the remaining $n = m - t$ points form the set of customers, (3) 10% of the m potential facility locations are selected randomly to form I_2, and (4) customer preference toward facilities is generated using the procedure described in Cánovas et al. (2007). With this procedure, ten instances of the following sizes $n \times m$ were generated: 225 × 25, 450 × 50, 675 × 75, 900 × 100, 1350 × 150, and 1800 × 200. A different coverage radius was used when generating each set of instances: 0.80, 0.70, 0.50, 0.30, 0.25, and 0.20, respectively. The p parameter values were set to 3, 6, 10, 13, 20, and 27 for instances of size 225 × 25, 450 × 50, 675 × 75, 900 × 100, 1350 × 150, and 1800 × 200, respectively.

To evaluate the quality of the solutions obtained by the proposed heuristics, the single-level reformulation was coded and run using the FICO XPRESS 7.8 commercial software. Both heuristics (the GRASP heuristic and hybrid GRASP-Tabu heuristic) were coded in C++. The single-level reformulation and both heuristics were run on an Intel Xenon(R) processor at 3.10 GHz with 8GB of RAM. The computational experimentation for both heuristics and the obtained results are discussed below.

The FICO XPRESS Optimization Suite provides a mathematical programming framework with different algorithms to solve LP problems and mixed integer LP problems (MIP). For LP problems, it

Algorithm 3.2 Local search procedure.

function LocalSearch($SolValue$)
  $Best \leftarrow SolValue$
  $Stop \leftarrow$ **false**
  repeat
    for all $i_2 \in S$ and $i_1 \in I_1 \setminus S$ **do**
      for all $j \in J$ **do**

$$\delta_j^{i_1,i_2} = \begin{cases} -D_j, & \text{if } A(j) = i_2 \text{ and } (j \notin \cup_{i \in S \setminus \{i_2\} \cup \{i_1\}} J(i) \text{ or } \arg\max_{i \in (S \cup I_2) \setminus \{i_2\} \cup \{i_1\}} \{g_{ij}\} \in I_2) \\ D_j, & \text{if } j \in J(i_1) \text{ and } (j \notin \cup_{i \in (S \cup I_2)} J(i)) \text{ or } (A(j) \in I_2 \text{ and } g_{A(j),j} < g_{i_1,j})) \\ 0, & \text{otherwise} \end{cases}$$

      end for
      $\Delta_{i_1,i_2} \leftarrow 0$
      for all $j \in J$ **do**
        $\Delta_{i_1,i_2} \leftarrow \Delta_{i_1,i_2} + \delta_j^{i_i,i_2}$
      end for
    end for
    $\Delta \leftarrow \max\{\Delta_{i_1,i_2} : i_2 \in S \text{ and } i_1 \in I_1 \setminus S\}$
    if $\Delta > 0$ **then**
      $(i_1^*, i_2^*) \in \arg\max_{i_2 \in S, i_1 \in I_1 \setminus S}\{\Delta_{i_1,i_2}\}$
      $S \leftarrow S \cup \{i_1^*\} \setminus \{i_2^*\}$
      $Best \leftarrow Best + \Delta$
      for all $j \in J$ **do**
        $A(j) \leftarrow \arg\max_{i \in (S \cup I_2) \cap I(j)} \{g_{ij}\}$
      end for
    else
      $Stop \leftarrow$ **true**
    end if
  until $Stop$
  return $Best$
end function

Algorithm 3.3 GRASP heuristic.

function GRASP($NumberOfIterations, \alpha_{initial}, \alpha_{final}, ItersFixed, Increment$)
  $IterCount \leftarrow 1$
  $\alpha \leftarrow \alpha_{initial}$
  $Best \leftarrow 0$
  while $(IterCount \leq NumberOfIterations)$ **do**
    if $(IterCount$ **mod** $Iterfixed = 0)$ **then**
      if $(\alpha = \alpha_{final})$ **then**
        $\alpha \leftarrow \alpha_{initial}$
      else
        $\alpha \leftarrow \alpha + Increment$
      end if
    end if
    $SolValue \leftarrow GreedyRandomized(\alpha)$
    $SolValue \leftarrow LocalSearch(SolValue)$
    if $(SolValue > Best)$ **then**
      $Best \leftarrow SolValue$
      $IterCount \leftarrow 0$
    end if
    $IterCount \leftarrow IterCount + 1$
  end while
  return $Best$
end function

includes the primal simplex, dual simplex, and Newton barrier algorithms. For MIP problems, the solver provides a powerful branch and bound framework that uses heuristic methods to quickly determine good solutions and cutting planes to strengthen the LP relaxations. The use of heuristics and cutting planes to provide primal and dual bounds might allow considerable reduction of enumerative effort. In addition, the MIP solver uses pre-solve procedures that might help reduce the problem matrix and in turn the dimension of the problem, thereby making it easier to solve. In preliminary tests with FICO XPRESS, it was observed that, for the largest instances, executing single-level reformulation for each instance could take more than 15 h. In addition, in preliminary tests, it was observed that executing the proposed heuristics for each instance never exceeded 250 s. Thus, a time limit of three hours (two orders of magnitude larger than the time required by the heuristics for the largest instances) was set to run the single-level reformulation in FICO XPRESS.

Here, we describe an evaluation of the proposed GRASP heuristic. The GRASP heuristic has two parameters that need to be adjusted: the α value and the termination criterion (number of iterations without improvement). The selection of the α parameter in the construction phase of a GRASP heuristic may affect the solution quality; therefore, it is very important to select an appropriate value for α. However, due to difficulty in setting a value for α that is suitable for all instances of a given problem, several options have been used in previous works. A trial-and-error strategy is often used to find the most suitable parameter value for each data instance. Other implementations (Delmaire et al., 1999; Prais & Ribeiro, 2000; Ríos-Mercado & Fernández, 2009) automatically tune the α value using a reactive procedure. These procedures are usually based on probabilities to select α values from a discrete set of values, i.e., the α value used in each iteration. For these procedures to work correctly, the heuristic algorithm must perform

Algorithm 3.4 Tabu search procedure.

function TabuSearch($SolValue$)
  $Best \leftarrow SolValue$
  $StopCriterion \leftarrow$ **false**
  repeat
    for all $i_2 \in S$ and $i_1 \in I_1 \setminus S$ **do**
      for all $j \in J$ **do**

$$\delta_j^{i_1,i_2} = \begin{cases} -D_j, & \text{if } A(j) = i_2 \text{ and } (j \notin \cup_{i \in S \setminus \{i_2\} \cup \{i_1\}} J(i) \text{ or } \arg\max_{i \in (S \cup I_2) \setminus \{i_2\} \cup \{i_1\}} \{g_{ij}\} \in I_2) \\ D_j, & \text{if } j \in J(i_1) \text{ and } (j \notin \cup_{i \in (S \cup I_2)} J(i)) \text{ or } (A(j) \in I_2 \text{ and } g_{A(j),j} < g_{i_1,j})) \\ 0, & \text{otherwise} \end{cases}$$

      end for
      $\Delta_{i_1,i_2} \leftarrow 0$
      for all $j \in J$ **do**
        $\Delta_{i_1,i_2} \leftarrow \Delta_{i_1,i_2} + \delta_j^{i_i,i_2}$
      end for
    end for
    $Candidates \leftarrow \left\{(i_1, i_2) : i_1 \in I_1 \setminus S, i_2 \in S \text{ and } (i_2 \notin TabuList \text{ or } SolValue + \Delta_{i_1,i_2} > Best)\right\}$
    $\Delta \leftarrow \max\{\Delta_{i_1,i_2} : (i_1, i_2) \in Candidates\}$
    $(i_1^*, i_2^*) \in \arg\max\{\Delta_{i_1,i_2} : (i_1, i_2 \in Candidates\}$
    $S \leftarrow S \cup \{i_1^*\} \setminus \{i_2^*\}$
    for all $j \in J$ **do**
      $A(j) \leftarrow \arg\max_{i \in (S \cup I_2) \cap I(j)} \{g_{ij}\}$
    end for
    Update $TabuList$
    $SolValue \leftarrow SolValue + \Delta$
    if $SolValue > Best$ **then**
      $Best \leftarrow SolValue$
    end if
    Update $StopCriterion$
  until (**not** $StopCriterion$)
  return $Best$
end function

many iterations. In this study, the solution quality of the proposed algorithms is sufficient without requiring many iterations; thus, a reactive procedure is not necessary. Instead, in all tests, the α parameter was varied from 0.75 to 0.95, starting at 0.75 and increasing α by 0.05 every five iterations.

The criterion used to terminate the GRASP heuristic is a fixed number of iterations without improvement. Two sets of tests were performed to evaluate the quality of the results obtained with the proposed GRASP heuristic and heuristic robustness. In the first set of tests, the termination criterion was 50 iterations without improvement, and in the second set of tests, the termination criterion was 100 iterations without improvement. Each set of tests consisted of running each test instance five times. To evaluate solution quality, the best solution from the five runs was compared to the optimum or best known solution of each instance, and, to evaluate heuristic robustness, the deviation between the best and worst solutions was measured.

Tables 1 and 2 describe the results obtained with the proposed GRASP heuristic. The first column shows the size of the instances, and the next three columns show the average percentage deviation of the best solutions obtained in each instance with respect to the optimal or best known solution (Avg. Best %), the average percentage deviation of the solution's mean with respect to the optimal or best known solution (Avg. Mean %), and the average percentage deviation of the worst solutions obtained with respect to the optimal or best known solution (Avg. Worst %). Finally, the last two columns show the average CPU time in seconds required by the GRASP heuristic and XPRESS, respectively.

Table 1
GRASP heuristic results (50 iterations).

Size	Avg. Best %	Avg. Mean %	Avg. Worst %	CPU (sec)	XPRESS CPU(sec)
225 × 25	0.000	0.000	0.000	0.0	1.9
450 × 50	0.000	0.001	0.005	0.0	6.4
675 × 75	0.000	0.018	0.031	1.5	21.1
900 × 100	0.000	0.002	0.010	4.5	65.3
1350 × 150	0.000	0.004	0.018	24.1	2562.6
1800 × 200	0.000	0.011	0.032	79.1	> 10,800.0

Table 2
GRASP heuristic results (100 iterations).

Size	Avg. Best %	Avg. Mean %	Avg. Worst %	CPU (sec)	XPRESS CPU (sec)
225 × 25	0.000	0.000	0.000	0.0	1.9
450 × 50	0.000	0.000	0.000	0.2	6.4
675 × 75	0.000	0.018	0.031	3.1	21.1
900 × 100	0.000	0.002	0.010	9.0	65.3
1350 × 150	0.000	0.001	0.003	44.3	2562.6
1800 × 200	0.000	0.005	0.015	142.6	> 10,800.0

Note that, for instances of size 1800 × 200, XPRESS was not able to find the optimal solution in the three-hour time limit for nine out of the ten instances. In addition, for these nine instances, the best feasible solution value obtained by XPRESS was worse than the solution obtained by the GRASP heuristic. In both sets of tests, the algorithm found the optimal or best known solution

Algorithm 3.5 Hybrid GRASP-Tabu heuristic.

function HYBRID($NumberOfIterations$, $\alpha_{initial}$, α_{final}, $ItersFixed$, $Increment$)
  $IterCount \leftarrow 1$
  $\alpha \leftarrow \alpha_{initial}$
  $Best \leftarrow 0$
  while ($IterCount \leq NumberOfIterations$) **do**
    if ($IterCount$ **mod** $Iterfixed = 0$) **then**
      if ($\alpha = \alpha_{final}$) **then**
        $\alpha \leftarrow \alpha_{initial}$
      else
        $\alpha \leftarrow \alpha + Increment$
      end if
    end if
    $SolValue \leftarrow GreedyRandomized(\alpha)$
    $SolValue \leftarrow TabuSearch(SolValue)$
    if ($SolValue > Best$) **then**
      $Best \leftarrow SolValue$
      $IterCount \leftarrow 0$
    end if
    $IterCount \leftarrow IterCount + 1$
  end while
  return $Best$
end function

Table 3
Hybrid GRASP-Tabu results.

Size	Avg. Best %	Avg. Mean %	Avg. Worst %	CPU (sec)	XPRESS CPU (sec)
225 × 25	0.000	0.000	0.000	0.0	1.9
450 × 50	0.000	0.000	0.000	1.4	6.4
675 × 75	0.000	0.006	0.031	7.6	21.1
900 × 100	0.000	0.000	0.000	13.9	65.3
1350 × 150	0.000	0.000	0.000	58.8	2562.6
1800 × 200	0.000	0.001	0.001	172.8	> 10,800.0

Table 4
Detailed hybrid GRASP-Tabu results.

Name	Optimal or Best Known	Best %	Avg. %	Worst %	CPU
mcb225 × 25-01	20,870	0.000	0.000	0.000	< 1.00
mcb225 × 25-02	18,562	0.000	0.000	0.000	< 1.00
mcb225 × 25-03	19,546	0.000	0.000	0.000	< 1.00
mcb225 × 25-04	21,236	0.000	0.000	0.000	< 1.00
mcb225 × 25-05	20,160	0.000	0.000	0.000	< 1.00
mcb225 × 25-06	19,458	0.000	0.000	0.000	< 1.00
mcb225 × 25-07	20,243	0.000	0.000	0.000	< 1.00
mcb225 × 25-08	20,511	0.000	0.000	0.000	< 1.00
mcb225 × 25-09	19,815	0.000	0.000	0.000	< 1.00
mcb225 × 25-10	20,210	0.000	0.000	0.000	< 1.00
mcb450 × 50-11	73,623	0.000	0.000	0.000	1.80
mcb450 × 50-12	79,903	0.000	0.000	0.000	1.20
mcb450 × 50-13	76,999	0.000	0.000	0.000	2.20
mcb450 × 50-14	68,361	0.000	0.000	0.000	1.20
mcb450 × 50-15	73,454	0.000	0.000	0.000	1.60
mcb450 × 50-16	72,540	0.000	0.000	0.000	1.20
mcb450 × 50-17	79,528	0.000	0.000	0.000	1.20
mcb450 × 50-18	75,316	0.000	0.000	0.000	1.20
mcb450 × 50-19	74,286	0.000	0.000	0.000	1.00
mcb450 × 50-20	80,620	0.000	0.000	0.000	1.20
mcb675 × 75-21	182,743	0.000	0.000	0.000	7.40
mcb675 × 75-22	176,540	0.000	0.061	0.306	10.40
mcb675 × 75-23	183,008	0.000	0.000	0.000	7.00
mcb675 × 75-24	177,188	0.000	0.000	0.000	8.20
mcb675 × 75-25	185,892	0.000	0.000	0.000	8.00
mcb675 × 75-26	176,562	0.000	0.000	0.000	6.00
mcb675 × 75-27	201,912	0.000	0.000	0.000	6.80
mcb675 × 75-28	188,272	0.000	0.000	0.000	6.20
mcb675 × 75-29	203,018	0.000	0.000	0.000	8.80
mcb675 × 75-30	183,625	0.000	0.000	0.000	7.20
mcb900 × 100-31	300,826	0.000	0.000	0.000	12.60
mcb900 × 100-32	325,846	0.000	0.000	0.000	12.60
mcb900 × 100-33	328,118	0.000	0.000	0.000	15.00
mcb900 × 100-34	309,788	0.000	0.000	0.000	11.60
mcb900 × 100-35	318,625	0.000	0.000	0.000	15.20
mcb900 × 100-36	319,178	0.000	0.000	0.000	13.40
mcb900 × 100-37	314,034	0.000	0.000	0.000	15.80
mcb900 × 100-38	327,188	0.000	0.000	0.000	12.00
mcb900 × 100-39	317,017	0.000	0.000	0.000	17.80
mcb900 × 100-40	298,779	0.000	0.000	0.000	13.20
mcb1350 × 150-41	693,053	0.000	0.000	0.000	59.20
mcb1350 × 150-42	723,185	0.000	0.000	0.000	65.80
mcb1350 × 150-43	742,178	0.000	0.000	0.000	61.20
mcb1350 × 150-44	688,609	0.000	0.000	0.000	66.00
mcb1350 × 150-45	684,277	0.000	0.000	0.000	61.80
mcb1350 × 150-46	695,860	0.000	0.000	0.000	44.00
mcb1350 × 150-47	690,537	0.000	0.000	0.000	48.60
mcb1350 × 150-48	693,398	0.000	0.000	0.000	61.60
mcb1350 × 150-49	731,687	0.000	0.000	0.000	50.60
mcb1350 × 150-50	723,746	0.000	0.000	0.000	69.00
mcb1800 × 200-51	1,281,284	0.000	0.000	0.000	144.40
mcb1800 × 200-52	1,356,246	0.000	0.006	0.010	211.80
mcb1800 × 200-53	1,258,479	0.000	0.000	0.000	163.80
mcb1800 × 200-54	1,314,660	0.000	0.001	0.004	168.60
mcb1800 × 200-55	1,300,838	0.000	0.000	0.000	145.20
mcb1800 × 200-56	1,256,017	0.000	0.000	0.000	157.60
mcb1800 × 200-57	1,290,808	0.000	0.000	0.000	206.20
mcb1800 × 200-58	1,280,134	0.000	0.000	0.000	180.20
mcb1800 × 200-59	1,287,646	0.000	0.000	0.000	201.00
mcb1800 × 200-60	1,331,171	0.000	0.000	0.000	149.20

for all instances in at least one out of the five runs. Moreover, the average percentage deviation of the worst solutions obtained with respect to the optimal or best known solution never exceeded 0.032% for the experiment with termination criterion of 50 iterations without improvement and 0.031% for the experiment with 100 iterations. Clearly, the algorithm is more robust by increasing the number of iterations without improvement in the termination criterion given that the deviation between the best and the worst solutions is reduced.

Table 3 shows the results of the proposed hybrid GRASP-Tabu heuristic. As with the GRASP heuristic tests, the α parameter was varied from 0.75 to 0.95, starting at 0.75 and increasing by 0.05 every five iterations. In addition, for this test, the number of iterations without improvement was set to 50, and Tabu tenure was set to seven iterations. Again, the test consisted of five runs for each test instance. To evaluate the solution quality, the best solution from the five runs was compared with the optimum or best known solution of each instance, and, to evaluate heuristic robustness, the deviation between the best and worst solutions was measured. Clearly, the hybrid GRASP-Tabu heuristic gives the best quality results. The average percentage deviation of the solution's mean with respect to the optimal or best known solution in the worst case was 0.006%, which is much lower compared with the 0.018% of the GRASP heuristic. In addition, the average percentage deviation of the worst solutions obtained with respect to the optimal or best known solution was improved substantially for instances greater than 675 × 75. In only three out of the 60 instances, the algorithm did not find the optimal or best known solution in the five runs, clearly showing that the algorithm is robust. Finally, although the average CPU time was greater for the hybrid GRASP-Tabu heuristic, the average time only increased slightly than the CPU time of the GRASP heuristic and remained much lower than the CPU time of the mathematical model run in FICO XPRESS.

The detailed results for each instance are shown in Table 4. The first column shows the name of each instance, and the second column shows the optimal or best known solution. The following three columns show the best, average, and worst percentage deviations of the solutions with respect to the optimal or best known solution. The last column shows the average CPU time in seconds.

5. Conclusions

In this study, the maximal covering location problem with customer preference ordering was investigated. Two different models for the problem were presented, i.e., a bilevel model and a single-level equivalent model. The equivalence between these two models was also discussed. Two heuristic algorithms were proposed to obtain lower bounds to the optimal solution of the problem: a GRASP heuristic and a hybrid GRASP-Tabu heuristic that replaces the local search phase of the GRASP heuristic with a Tabu search procedure.

Both algorithms were tested with a set of 60 randomly generated instances, and several computational tests were performed to evaluate the proposed heuristics. First, the single-level reformulation was solved for all instances using mathematical programming software (FICO XPRESS) with two purposes: (1) to evaluate the quality of the solutions obtained by the proposed heuristics and (2) to evaluate the efficiency of an exact method as instance size increases.

The heuristics solutions were compared with solutions of the single-level reformulation of the problem. According to the results of the computational tests, the two proposed heuristics provided good quality solutions. In 51 of the 60 test instances, the proposed heuristics found the optimal solution in at least one of the five executions. In addition, for the remaining nine instances, for which the optimality of the solutions could not be verified within the three-hour time limit using FICO XPRESS, the heuristics provided better lower bounds than those obtained with FICO XPRESS. However, the proposed GRASP-Tabu hybrid heuristic outperformed the GRASP heuristic because it is more robust without significantly increasing CPU time.

The results indicate that the enumerative effort required by FICO XPRESS increases considerably as instance size grows. In preliminary tests, for the largest instances, FICO XPRESS required more than 15 h. Note that, in the worst case, the time required by the proposed heuristic never exceeds 250 s, thereby demonstrating the efficiency of the proposed heuristics.

Metaheuristic methods and cutting-plane methods are very useful to design special purpose exact methods. Metaheuristics can provide high-quality incumbents and primal bounds in Branch & Bound methods, and cutting-plane methods improve dual bounds. Together, they might reduce the enumerative effort of an exact method. A future direction of this research can be the combination of heuristics and cutting planes for the design of an exact method for the maximal covering location problem with customer preference ordering.

Another future research direction can be to consider heuristic methods for the bilevel formulation of the problem based on a two-player, two-stage strategic game with perfect information, similar to the ones proposed in Robbins and Lunday (2016).

References

Abdinnour-Helm, S., & Hadley, S. (2000). Tabu search based heuristic for multi-floor facility layout. *International Journal of Production Research, 38*(2), 365–383.
Bard, J. (1998). *Practical bilevel optimization: Applications and algorithms*. Kluwer Academic Publishers.
Belotti, P., Labbé, M., Maffioli, F., & Ndiaye, M. (2007). A branch-and-cut method for the obnoxious *p*-median problem. *4OR, 5*(4), 299–314.
Blum, C. (2010). Hybrid metaheuristics. *Computers & Operations Research, 37*(3), 430–431.
Camacho-Vallejo, J., Cordero-Franco, A., & González-Ramírez, R. (2014). Solving the bilevel facility location problem under preferences by a Stackelberg-evolutionary algorithm. *Mathematical Problems in Engineering, 2014*, 1–14.
Cánovas, L., García, S., Labbé, M., & Marín, A. (2007). A strengthened formulation for the simple plant location problem with order. *Operations Research Letters, 35*(2), 141–150.
Church, R., & ReVelle, C. (1974). The maximal covering location problem. *Papers in Regional Science, 32*(1), 101–118.
Colomé, R., & Serra, D. (2001). Consumer choice in competitive location models: Formulations and heuristics. *Papers in Regional Science, 80*(4), 439–464.
Colson, B., Marcotte, P., & Savard, G. (2007). An overview of bilevel optimization. *Annals of Operations Research, 153*(1), 235–265.
Delmaire, H., Díaz, J. A., Fernández, E., & Ortega, M. (1999). Reactive GRASP and tabu search based heuristics for the single source capacitated plant location problem. *INFOR: Information Systems and Operational Research, 37*(3), 194–225.
Díaz, J. A., Luna, D. E., & Zetina, C. A. (2013). A hybrid algorithm for the manufacturing cell formation problem. *Journal of Heuristics, 19*(1), 77–96.
Duarte, A., & Martí, R. (2007). Tabu search and GRASP for the maximum diversity problem. *European Journal of Operational Research, 178*(1), 71–84.
Feo, T., & Resende, M. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization, 6*(2), 109–133.
Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research, 13*(5), 533–549.
Glover, F. (1989). Tabu search - part I. *ORSA Journal of Computing, 1*(3), 190–206.
Glover, F. (1990). Tabu search part II. *ORSA Journal on Computing, 2*(1), 4–32.
Hakimi, S. L. (1983). On locating new facilities in a competitive environment. *European Journal of Operational Research, 12*(1), 29–35.
Hanjoul, P., & Thill, J. C. (1987). Elements on planar analysis in spatial competition. *Regional Science and Urban Economics, 17*(3), 423–439.
Hansen, P., Jaumard, B., & Savard, G. (1992). New branch and bound rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing, 13*(5), 1194–1217.
Hansen, P., Kochetov, Y., & Mladenovi, N. (2004). *Lower bounds for the uncapacitated facility location problem with user preferences*. Groupe d'études et de recherche en analyse des décisions, HEC Montréal.
Jeroslow, R. (1985). The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming, 32*(2), 146–164.
Kalashnikov, V., Dempe, S., Pérez-Valdéz, G., Kalashnikova, V., & Camacho–Vallejo, J. (2004). Bilevel programming and applications. *Mathematical Problems in Engineering, 31*, 1515–1526.
Kress, D., & Pesch, E. (2012). Sequential competitive location on networks. *European Journal of Operational Research, 217*(3), 483–499.
Laguna, M., & González-Velarde, J. L. (1991). A search heuristic for just-in-time scheduling in parallel machines. *Journal of Intelligent Manufacturing, 2*(4), 253–260.
Lee, J. M., & Lee, Y. H. (2012). Facility location and scale decision problem with customer preference. *Computers & Industrial Engineering, 63*(1), 184–191.
Lim, A., & Wang, F. (2004). A smoothed dynamic tabu search embedded GRASP for m-VRPTW. In *Tools with artificial intelligence, 2004. ICTAI 2004. 16th IEEE international conference on* (pp. 704–708).
Lozano, M., & García-Martínez, C. (2010). Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. *Computers & Operations Research, 37*(3), 481–497.
Marić, M., Stanimirović, Z., & Milenković, N. (2012). Metaheuristic methods for solving the bilevel uncapacitated facility location problem with clients' preferences. *Electronic Notes in Discrete Mathematics, 39*, 43–50.
Ognjanović, I., GaševiÄ, D., & Bagheri, E. (2013). A stratified framework for handling conditional preferences: An extension of the analytic hierarchy process. *Expert Systems with Applications, 40*(4), 1094–1115.
Plastria, F. (2001). Static competitive facility location: An overview of optimisation approaches. *European Journal of Operational Research, 129*(3), 461–470.
Prais, M., & Ribeiro, C. (2000). Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment. *INFORMS Journal on Computing, 12*(3), 164–176. doi:10.1287/ijoc.12.3.164.12639.
Resende, M. G. C. (1998). Computing approximate solutions of the maximum covering problem with GRASP. *Journal of Heuristics, 4*(2), 161–177.
Resende, M. G. C., & González-Velarde, J. L. (2003). GRASP: procedimientos de busqueda miopes aleatorizados y adaptativos. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial, 7*(19), 61–76.
Ríos-Mercado, E., & Fernández, R. (2009). A reactive {GRASP} for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research, 36*(3), 755–776.
Robbins, M., & Lunday, B. (2016). A bilevel formulation of the pediatric vaccine pricing problem. *European Journal of Operational Research, 264*, 634–645.
Russell, S., & Norvig, P. (2010). *Artifitial intelligence: A modern approach*. New Jersey, United States: Prentice Hall.
Talbi, E. G. (2002). A taxonomy of hybrid metaheuristics. *Journal of Heuristics, 8*(5), 541–564.
Vasil'ev, I. L., Klimentova, K. B., & Kochetov, Y. A. (2009). New lower bounds for the facility location problem with clients' preferences. *Computational Mathematics and Mathematical Physics, 49*(6), 1010–1020.
Vasilyev, I. L., & Klimentova, K. B. (2010). The branch and cut method for the facility location problem with client's preferences. *Journal of Applied and Industrial Mathematics, 4*(3), 441–454.