The Maximal Covering Location Bi-level Problem

José-Fernando Camacho-Vallejo

Facultad de Ciencias Físico-Matemáticas
Universidad Autónoma de Nuevo León, México
jose.camachovl@uanl.edu.mx

VI International Workshop on Locational Analysis and Related Problems
November 25th, 2015

Co-authors

Martha-Selene Casas-Ramírez
Facultad de Ciencias Físico Matemáticas
Universidad Autónoma de Nuevo León, México
martha.casasrm@uanl.edu.mx

Juan A. Díaz
Departamento de Actuaría, Física y Matemáticas
Universidad de las Américas Puebla, México
juana.diaz@udlap.mx

Dolores E. Luna
Departamento de Ingenierá Industrial y Mecánica
Universidad de las Américas Puebla, México
dolorese.luna@udlap.mx

Table of contents

1. Introduction
2. Mathematical model
3. Single-level reformulations
4. Genetic algorithm
5. Parameter tuning
6. Numerical results
7. Conclusions and further research
8. References
9. Announcements

Description of the situation studied

We consider a situation when a firm is already operating in the market and a new competitor will enter in the same market

The original firm has some facilities covering a percentage of the customers.

Since there is only one firm controlling the market, it does not matter customers preferences because they do not have other options.

Description of the situation studied

Then, a competing firm wants to locate p facilities for capturing the demand of some portion of the market.

Now the customers can choose the facility of their preference. That is, both firms do not have control regarding the allocation decision.

It is assumed that the original firm already has a number of existing facilities and will not locate new ones.

The competing firm will enter to the market knowing the existing locations.

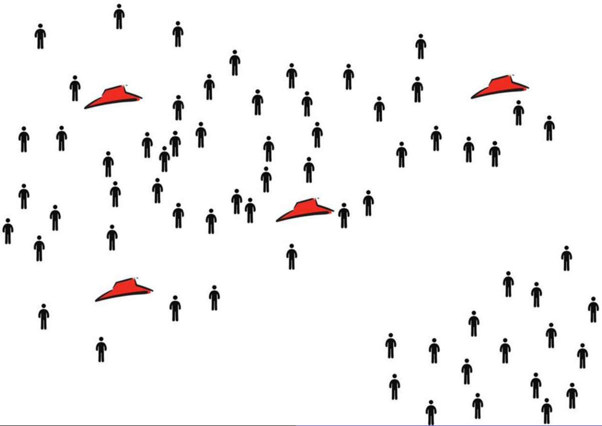
Description of the situation studied

This proposed problem can be seen as a variant of the $(r|X_p)$-medianoid problem.

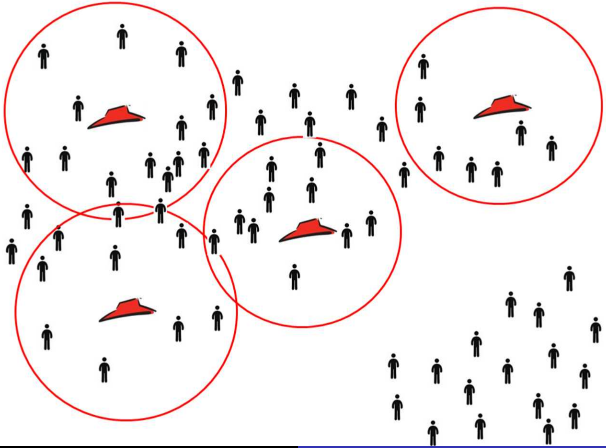
The main difference is that here, we are considering that the customers will decide their allocation based on their preferences towards the facilities -old or new ones-.

To the best of our knowledge, this is the first paper that handles a covering problem considering the customers' preferences. The bi-level model is a natural way to formulate the problem here addressed.

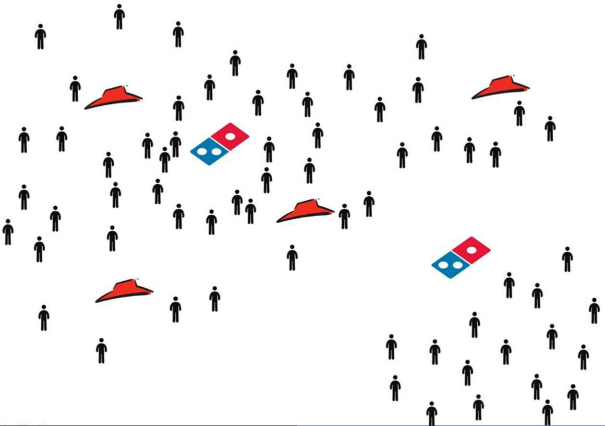
Illustrative example



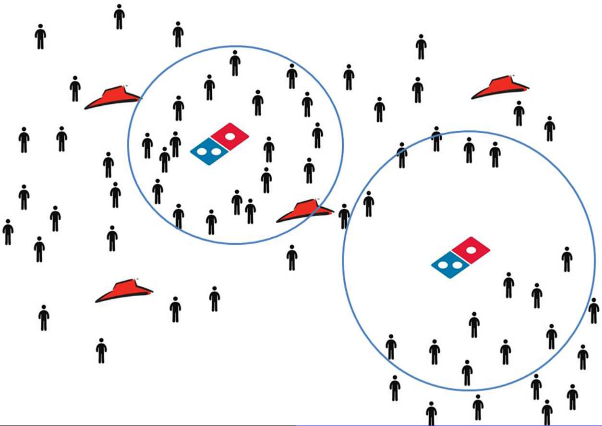
Illustrative example



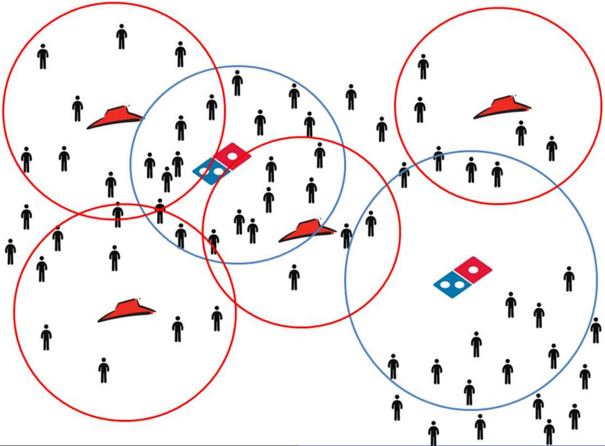
Illustrative example



Illustrative example



Illustrative example



Related literature

Different criteria can be used to simulate customer's behaviour. After an intensive literature review, only in Lee & Lee (2012) the customers' preferences are considered for covering problems.

The problem maximizes the weighted covered demand with respect to customers' preferences, this is maximizes the number of customers served by the facility multiplied by their preference index.

They consider a decision maker who controls both variables (allocation and location) and he intends to give an integrated solution bearing in mind the preferences of the customers.

On the other hand, if the customers can decide their own allocation based on the located facilities, then the problem is approached from a quite different point of view.

Related literature

In the latter situation, there are two decision makers involved in the decision process and a hierarchy exists among them; hence, bi-level programming is a suitable approach.

Customers' preferences have been considered for other location problems, such as, the uncapacitated FLP. In this problem, the upper level concerns about the costs minimization (location and distribution); while the lower level aims to optimize the customer's preferences.

The papers related to this problem refers to papers devoted to solve single-level reformulations of the bi-level problem are: Hansen et al. (2004), Vasil'ev et al. (2009), Vasil'ev & Klimentova (2010)

Also, papers that developed metaheuristics for solving the bi-level version of the UFLBP are: Maric et al. (2012) and Camacho-Vallejo et al. (2014).

Problem's statement

Consider the situation when a set of customers has a demand associated to each of them. Some customers are covered by existing facilities in the market.

Then, a competitor wants to enter the market by opening some new facilities. Now, as the customers have new options, they are free to select the facility that will serve them.

This situation is formulated as a bi-level programming problem, where the leader is the new competitor and the follower is the set of customers.

Problem's statement

The leader will decide the location of a limited number of facilities aiming to maximize the demand covered. Whereas, the follower will allocate the customers to their most preferred facility seeking to maximize customer's preferences.

It is assumed that already existing facilities will remain opened.

Customers are allocated to the most preferred facility within the coverage radius, regardless of whether they are new or existing facilities.

Formulation

Let I_1 be the set of potential facilities and I_2 be the set of existing ones, where $I = I_1 \cup I_2$.

Let J_1 be the set of customers covered by I_2 and J_2 be the set of customers uncovered by I_2, where $J = J_1 \cup J_2$.

Additionally, we define, for each $j \in J$ the set $I(j)$ which contains the set of facilities that cover customer j and, for each $i \in I$, $J(i)$ denote the customer set that is covered by location i.

Let D_j be the demand associated to the customer $j \in J$.

Formulation

Consider p as the number of facilities that the leader will open for entering in the market; and, let g_{ij} be the preference of the customer $j \in J$ towards the facility $i \in I$.

The leader's binary decision variables are denoted by y_i, which indicate whether the facility $i \in I_1$ is opened or not.

The follower's binary decision variables, denoted by x_{ij}, express whether the customer $j \in J$ is allocated to the facility $i \in I$ or not.

For identifying sets I(j) and J(i) consider $d_{ij} \leq r$, where d_{ij} is the distance from customer $j \in J$ to facility $i \in I$ and r is a predetermined coverage ratio.

Formulation

$$\max_{y} \quad \sum_{i \in I_1} \sum_{j \in J(i)} D_j x_{ij} \tag{2.1}$$

$$\text{subject to:} \quad y_i = 1 \quad \forall i \in I_2 \tag{2.2}$$

$$\sum_{i \in I_1} y_i = p \tag{2.3}$$

$$y_i \in \{0, 1\} \quad \forall i \in I_1 \tag{2.4}$$

$$\text{where} \quad x \quad \text{solves} \tag{2.5}$$

$$\max_{x} \quad \sum_{i \in I} \sum_{j \in J(i)} g_{ij} x_{ij} \tag{2.6}$$

$$\text{subject to:} \quad \sum_{i \in I(j)} x_{ij} = 1 \quad \forall j \in J_1 \tag{2.7}$$

$$x_{ij} \leq y_i \quad \forall i \in I_1, j \in J(i) \tag{2.8}$$

$$\sum_{i \in I(j)} x_{ij} \leq 1 \quad \forall j \in J_2 \tag{2.9}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J(i) \tag{2.10}$$

Assumptions

Customers' preferences are assumed to be consecutive integer numbers from 1 to the total number of locations ($|I|$).

In order to have a well-defined bi-level problem it is important to carefully analyze the existence and uniqueness of the lower level's optimal solution.

Fortunately, the follower's problem has been studied before highlighting the following fact: if for each customer $j \in J$ the customer's preferences are positive consecutive numbers from 1 to $|I|$; then the optimal solution for the follower's problem will be unique.

The proof of this important result is given in Vasil'ev et al. (2009).

Follower's dual problem

The reformulation is made following the classical approach of using the primal-dual relationships for the follower's problem.

For guaranteeing optimality in the follower's decision two schemes are commonly considered:
i) force the equality of the objective functions of both primal and dual problems.
ii) include the complementarity slackness constraints.

These reformulations are possible because the linear relaxation of the lower level problem gives an optimal integer solution.

Follower's dual problem

Let α_j for all $j \in J_1$, β_{ij} for all $i \in I_1, j \in J(i)$ and γ_j for all $j \in J_2$ be the dual variables associated with the linear relaxation of the follower's problem.

Follower's dual problem

Let α_j for all $j \in J_1$, β_{ij} for all $i \in I_1, j \in J(i)$ and γ_j for all $j \in J_2$ be the dual variables associated with the linear relaxation of the follower's problem.

Then, the corresponding dual problem is as follows:

$$\max_{\alpha,\beta,\gamma} \quad \sum_{j\in J_1} \alpha_j + \sum_{i\in I_1} \sum_{j\in J(i)} y_i\beta_{ij} + \sum_{j\in J_2} \gamma_j \tag{3.1}$$

$$\alpha_j + \beta_{ij} \geq g_{ij} \quad \forall j \in J_1, i \in I_1 \cap I(j) \tag{3.2}$$

$$\beta_{ij} + \gamma_j \geq g_{ij} \quad \forall j \in J_2, i \in I_1 \cap I(j) \tag{3.3}$$

$$\alpha_j \geq g_{ij} \quad \forall j \in J_1, i \in I_2 \cap I(j) \tag{3.4}$$

$$\gamma_j \geq g_{ij} \quad \forall j \in J_2, i \in I_2 \cap I(j) \tag{3.5}$$

$$\beta_{ij} \geq 0 \quad \forall i \in I_1, j \in J(i) \tag{3.6}$$

$$\gamma_j \geq 0 \quad \forall j \in J_2 \tag{3.7}$$

Equality of the objective functions

It is easy to see that the second term in the dual's objective function is non-linear. Hence, an auxiliary variable is introduced for linearizing it. Let $\delta_{ij} = y_i \beta_{ij}$ for all $i \in I_1, j \in J(i)$.

It is worthy to note that, since y_i only can take the values 0 or 1, then $\delta_{ij} = \beta_{ij}$ when $y_i = 1$ and $\delta_{ij} = 0$ when $y_i = 0$.

Equality of the objective functions

It is easy to see that the second term in the dual's objective function is non-linear. Hence, an auxiliary variable is introduced for linearizing it. Let $\delta_{ij} = y_i \beta_{ij}$ for all $i \in I_1, j \in J(i)$.

It is worthy to note that, since y_i only can take the values 0 or 1, then $\delta_{ij} = \beta_{ij}$ when $y_i = 1$ and $\delta_{ij} = 0$ when $y_i = 0$.

Therefore, the following constraints are added:

$$\delta_{ij} \geq 0 \qquad \forall i \in I_1, j \in J(i) \tag{3.8}$$

$$\delta_{ij} \leq My_i \qquad \forall i \in I_1, j \in J(i) \tag{3.9}$$

$$\delta_{ij} \leq \beta_{ij} \qquad \forall i \in I_1, j \in J(i) \tag{3.10}$$

$$\delta_{ij} + M \geq \beta_{ij} + My_i \qquad \forall i \in I_1, j \in J(i) \tag{3.11}$$

Equality of the objective functions

The resulting single-level MIP problem is as follows:

$$\max_{y,x,\alpha,\beta,\gamma,\delta} \quad \sum_{i\in I_1}\sum_{j\in J(i)} D_j x_{ij} \tag{3.12}$$

$$y_i = 1 \qquad \forall i \in I_2 \tag{3.13}$$

$$\sum_{i\in I_1} y_i = p \tag{3.14}$$

$$\sum_{i\in I(j)} x_{ij} = 1 \qquad \forall j \in J_1 \tag{3.15}$$

$$x_{ij} \leq y_i \qquad \forall i \in I_1, j \in J(i) \tag{3.16}$$

$$\sum_{i\in I(j)} x_{ij} \leq 1 \qquad \forall j \in J_2 \tag{3.17}$$

$$\alpha_j + \beta_{ij} \geq g_{ij} \qquad \forall j \in J_1, i \in I_1 \cap I(j) \tag{3.18}$$

$$\beta_{ij} + \gamma_j \geq g_{ij} \qquad \forall j \in J_2, i \in I_1 \cap I(j) \tag{3.19}$$

Equality of the objective functions

$$\alpha_j \geq g_{ij} \quad \forall j \in J_1, i \in I_2 \cap I(j) \tag{3.20}$$

$$\gamma_j \geq g_{ij} \quad \forall j \in J_2, i \in I_2 \cap I(j) \tag{3.21}$$

$$\sum_{i \in I, j \in J(i)} g_{ij} x_{ij} = \sum_{j \in J_1} \alpha_j + \sum_{i \in I_1} \sum_{j \in J(i)} \delta_{ij} + \sum_{j \in J_2} \gamma_j \tag{3.22}$$

$$\delta_{ij} \geq 0 \quad \forall i \in I_1, j \in J(i) \tag{3.23}$$

$$\delta_{ij} \leq M y_i \quad \forall i \in I_1, j \in J(i) \tag{3.24}$$

$$\delta_{ij} \leq \beta_{ij} \quad \forall i \in I_1, j \in J(i) \tag{3.25}$$

$$\delta_{ij} + M \geq \beta_{ij} + M y_i \quad \forall i \in I_1, j \in J(i) \tag{3.26}$$

$$y_i \in \{0, 1\} \quad \forall i \in I \tag{3.27}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J(i) \tag{3.28}$$

$$\beta_{ij} \geq 0 \quad \forall i \in I_1, j \in J(i) \tag{3.29}$$

$$\gamma_j \geq 0 \quad \forall j \in J_2 \tag{3.30}$$

Optimality slackness constraints

The second approach followed is based on the use of the optimality slackness constraints associated to both primal and dual follower's problems.

Optimality slackness constraints

The second approach followed is based on the use of the optimality slackness constraints associated to both primal and dual follower's problems.

The constraints included are:

$$
\begin{aligned}
(\alpha_j + \beta_{ij} - g_{ij})x_{ij} &= 0 \quad \forall j \in J_1, i \in I_1 \cap I(j) && (3.31)\\
(\beta_{ij} + \gamma_j - g_{ij})x_{ij} &= 0 \quad \forall j \in J_2, i \in I_1 \cap I(j) && (3.32)\\
(\alpha_j - g_{ij})x_{ij} &= 0 \quad \forall j \in J_1, i \in I_2 \cap I(j) && (3.33)\\
(\gamma_j - g_{ij})x_{ij} &= 0 \quad \forall j \in J_2, i \in I_2 \cap I(j) && (3.34)\\
(y_i - x_{ij})\beta_{ij} &= 0 \quad \forall J \in J, i \in I_1 \cap I(j) && (3.35)\\
\left(\sum_{i \in I(j)} x_{ij}\right)\gamma_j &= 0 \quad \forall j \in J_2 && (3.36)
\end{aligned}
$$

Optimality slackness constraints

The latter set of constraints can be substituted by the following equations:

$$\begin{aligned}
\alpha_j + \beta_{ij} - g_{ij} &\leq M(1 - x_{ij}) \qquad \forall j \in J_1, i \in I_1 \cap I(j) && (3.37)\\
\beta_{ij} + \gamma_j - g_{ij} &\leq M(1 - x_{ij}) \qquad \forall j \in J_2, i \in I_1 \cap I(j) && (3.38)\\
\alpha_j - g_{ij} &\leq M(1 - x_{ij}) \qquad \forall j in J_1, i \in I_2 \cap I(j) && (3.39)\\
\gamma_j - g_{ij} &\leq M(1 - x_{ij}) \qquad \forall j \in J_2, i \in I_2 \cap I(j) && (3.40)\\
\beta_{ij} &\leq M(1 - y_i + x_{ij}) \qquad \forall j \in J, i \in I_1 \cap I(j) && (3.41)\\
\gamma_j &\leq M\left(\sum_{i \in I \cap I(j)} x_{ij}\right) \qquad \forall j \in J_2 && (3.42)
\end{aligned}$$

where, as before, M is a sufficient large constant.

Optimality slackness constraints

Therefore, the second reformulation that arose from the described approach is the following single-level mixed integer linear programming problem:

$$\max_{y,x,\alpha,\beta,\gamma} \sum_{i\in I_1}\sum_{j\in J(i)} D_j x_{ij} \tag{3.43}$$

$$y_i = 1 \quad \forall i \in I_2 \tag{3.44}$$

$$\sum_{i\in I_1} y_i = p \tag{3.45}$$

$$\sum_{i\in I(j)} x_{ij} = 1 \quad \forall j \in J_1 \tag{3.46}$$

$$x_{ij} \leq y_i \quad \forall i \in I_1, j \in J(i) \tag{3.47}$$

$$\sum_{i\in I(j)} x_{ij} \leq 1 \quad \forall j \in J_2 \tag{3.48}$$

$$\alpha_j + \beta_{ij} \geq g_{ij} \quad \forall j \in J_1, i \in I_1 \cap I(j) \tag{3.49}$$

$$\beta_{ij} + \gamma_j \geq g_{ij} \quad \forall j \in J_2, i \in I_1 \cap I(j) \tag{3.50}$$

Optimality slackness constraints

$$
\begin{aligned}
\alpha_j &\geq g_{ij} \quad \forall j \in J_1, i \in I_2 \cap I(j) && (3.51)\\
\gamma_j &\geq g_{ij} \quad \forall j \in J_2, i \in I_2 \cap I(j) && (3.52)\\
\alpha_j + \beta_{ij} - g_{ij} &\leq M\,(1 - x_{ij}) \quad \forall j \in J_1, i \in I_1 \cap I(j) && (3.53)\\
\beta_{ij} + \gamma_j - g_{ij} &\leq M\,(1 - x_{ij}) \quad \forall j \in J_2, i \in I_1 \cap I(j) && (3.54)\\
\alpha_j - g_{ij} &\leq M\,(1 - x_{ij}) \quad \forall j \in J_1, i \in I_2 \cap I(j) && (3.55)\\
\gamma_j - g_{ij} &\leq M\,(1 - x_{ij}) \quad \forall j \in J_2, i \in I_2 \cap I(j) && (3.56)\\
\beta_{ij} &\leq M\,(1 - y_i + x_{ij}) \quad \forall j \in J, i \in I_1 \cap I(j) && (3.57)\\
\gamma_j &\leq M\left(\sum_{i \in I \cap I(j)} x_{ij}\right) \quad \forall j \in J_2 && (3.58)\\
y_i &\in \{0,1\} \quad \forall i \in I && (3.59)\\
x_{ij} &\in \{0,1\} \quad \forall i \in I, j \in J(i) && (3.60)\\
\beta_{ij} &\geq 0 \quad \forall i \in I_1, j \in J(i) && (3.61)\\
\gamma_j &\geq 0 \quad \forall j \in J_2 && (3.62)
\end{aligned}
$$

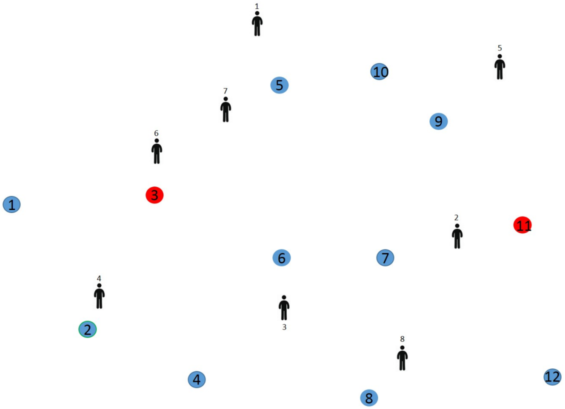
Solution coding

In a schematic way, we will consider that first the leader selects a set of facilities and then the lower level is optimally solved, by allocating customers to their most preferred facility.

The coding for a leader's solution consists in a vector of size p in which each position indicates the index of an opened facility; the indexes are ordered in an increasing way.

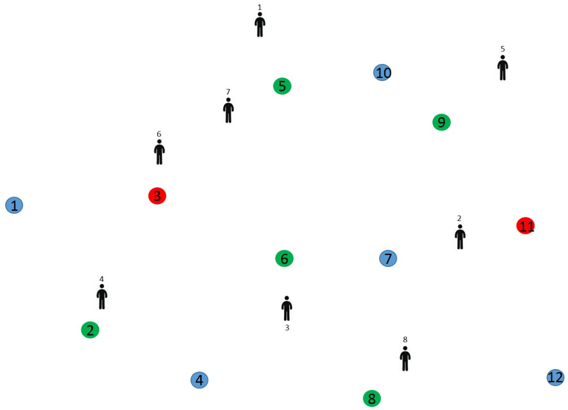
The follower's solution is codified as a vector of size $|J|$, that is, the total number of customers. In this vector, the value of the j^{th} position corresponds to the facility to which customer j^{th} is allocated.

Solution coding



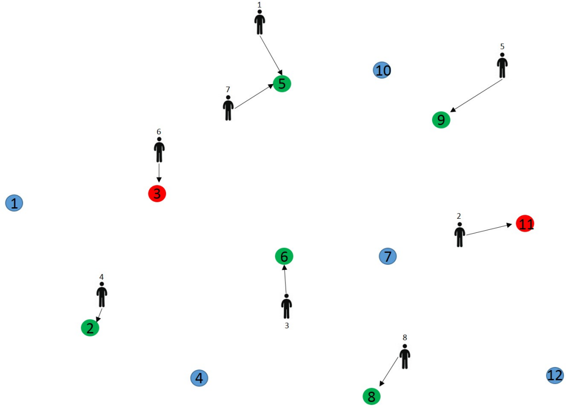
Solution coding

$$y = [2, 5, 6, 8, 9]$$



Solution coding

$$x = [5, 11, 6, 2, 9, 3, 5, 8]$$



Initial population

The initial set of solutions are created randomly but based on a specific distribution probability given by

$$q_i = \frac{\sum_{j \in J(i)} D_j}{\sum_{i \in I_1} \sum_{j \in J(i)} D_j} \tag{4.1}$$

for all $i \in I_1$; where $\sum_{j \in J(i)} D_j$ represents the demand covered by facility $i \in I_1$

The main objective is to favour those locations that might cover larger demands by assigning it a higher probability to be included in the solution.

Selection mechanism

One of the most common methods for avoiding premature convergence in the selection phase, is the tournament strategy.

For each solution of the population five tournaments are made against another solution randomly selected. In each tournament the leader's objective function values (fitness) are compared. The number of times that each solution wins the tournament is recorded.

After the tournaments are performed, an elitist selection is made.

Genetic operators

Crossover: Select two solutions from the initial population and randomly choose a single point within the solutions for dividing the solutions. Then, the part of the solution after the selected crossover point is interchanged with the corresponding part of the other solution. Abortions may appear in this operator.

Genetic operators

Crossover: Select two solutions from the initial population and randomly choose a single point within the solutions for dividing the solutions. Then, the part of the solution after the selected crossover point is interchanged with the corresponding part of the other solution. Abortions may appear in this operator.

Mutation: Consider a solution and randomly choose a component. Then, randomly selects a potential facility that is not considered in the current solution and include it in the mutated one. Only feasible solutions are generated in this operator.

Description of the parameters

Three main parameters may be identified in the proposed genetic algorithm: population size (P_{size}), number of generations (Gen) and the probability (P) in the genetic operators (for performing crossover or mutation).

Description of the parameters

Three main parameters may be identified in the proposed genetic algorithm: population size (P_{size}), number of generations (Gen) and the probability (P) in the genetic operators (for performing crossover or mutation).

Preliminary tests with the following different values for each parameter were conducted:
$P_{size} \rightarrow$ 100, 200, 500 and 1000.

Description of the parameters

Three main parameters may be identified in the proposed genetic algorithm: population size (P_{size}), number of generations (Gen) and the probability (P) in the genetic operators (for performing crossover or mutation).

Preliminary tests with the following different values for each parameter were conducted:
$P_{size} \rightarrow$ 100, 200, 500 and 1000.
$Gen \rightarrow 2|I|$ and $3|I|$.

Description of the parameters

Three main parameters may be identified in the proposed genetic algorithm: population size (P_{size}), number of generations (Gen) and the probability (P) in the genetic operators (for performing crossover or mutation).

Preliminary tests with the following different values for each parameter were conducted:
$P_{size} \rightarrow$ 100, 200, 500 and 1000.
$Gen \rightarrow 2|I|$ and $3|I|$.
$P \rightarrow$ 0.65, 0.75, 0.85 and 0.95.

Description of the parameters

For conducting a full factorial design, 4(2)(4)=32 runs are needed for each instance.

We created a set of 60 instances based on a procedure described in Resende (1998). It can be divided in two blocks -medium (M) and large (L) size-.

Cuadro: Description of the instances.

	Block M			Block L		
Customers	225	450	675	900	1350	1800
Facilities	25	50	75	100	150	200

Description of the parameters

We decided to consider only the half of those 60 instances for the parameter tuning. That is, the block M.

After have ran the preliminary tests, it was noticed that the behaviour of the algorithm it is very similar for the same subset of instances with the same size.

Hence, three instances from each size within the medium size block were randomly chosen 225x25-(01,08,10), 450x50-(13,16,18) and 675x75-(23,25,29).

Description of the parameters

Five runs of the algorithm for each one of the 32 parameter configurations were executed.

The following values were measured from each run: the best leader's objective function value reached, the average of the best value found within the five runs, and the average time required for solving each instance.

For all the possible configurations, the algorithm was capable of finding the optimal value for every instance but not in all the five runs.

Tuning the size of the population

Table 1. Average of objective value with $P_{size}=100$.

	225x25-01	225x25-08	225x25-10	450x50-13	450x50-16	450x50-18	675x75-23	675x75-25	675x75-29
Best	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(21,0.65)	20870.0	20511.0	20210.0	76959.0	72279.4	74445.2	181831.2	185188.4	201309.6
(21,0.75)	20870.0	20511.0	20210.0	76967.0	72303.6	75086.2	181059.4	185576.8	201886.2
(21,0.85)	20870.0	20511.0	20210.0	76881.0	72540.0	74992.8	182914.4	185343.0	202390.4
(21,0.95)	20870.0	20511.0	20210.0	76967.0	72366.8	75016.6	182888.0	185672.6	201946.0
(31,0.65)	20870.0	20511.0	20210.0	76959.0	72366.8	75000.8	181922.0	184751.4	201234.0
(31,0.75)	20870.0	20511.0	20210.0	76975.0	72212.4	74903.0	181917.8	185642.0	201320.6
(31,0.85)	20870.0	20511.0	20210.0	76959.0	72366.8	75070.4	182426.2	184999.2	201618.4
(31,0.95)	20870.0	20511.0	20210.0	76701.8	72540.0	75316.0	182680.0	185671.8	201669.0

Tuning the size of the population

Table 2. Average of objective value with $P_{size}=200$.

	225x25-01	225x25-08	225x25-10	450x50-13	450x50-16	450x50-18	675x75-23	675x75-25	675x75-29
Best	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(21,0.65)	20870.0	20511.0	20210.0	76999.0	72540.0	75244.8	182961.2	185762.0	202361.4
(21,0.75)	20870.0	20511.0	20210.0	76967.0	72540.0	75316.0	182681.4	185892.0	202145.6
(21,0.85)	20870.0	20511.0	20210.0	76975.0	72540.0	75316.0	182692.2	185892.0	202645.8
(21,0.95)	20870.0	20511.0	20210.0	76991.0	72540.0	75316.0	182747.2	185892.0	202513.6
(31,0.65)	20870.0	20511.0	20210.0	76991.0	72540.0	75316.0	182789.6	185802.6	202375.4
(31,0.75)	20870.0	20511.0	20210.0	76983.0	72485.0	75316.0	182867.6	185499.4	203018.0
(31,0.85)	20870.0	20511.0	20210.0	76983.0	72540.0	75316.0	182789.6	185617.4	202781.2
(31,0.95)	20870.0	20511.0	20210.0	76991.0	72540.0	75316.0	183008.0	185892.0	202662.2

Tuning the size of the population

Table 3. Average of objective value with $P_{size}=500$.

	225x25-01	225x25-08	225x25-10	450x50-13	450x50-16	450x50-18	675x75-23	675x75-25	675x75-29
Best	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(21,0.65)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(21,0.75)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(21,0.85)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(21,0.95)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(31,0.65)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(31,0.75)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(31,0.85)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(31,0.95)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0

Tuning the size of the population

Table 4. Average of objective value with P_{size}=*1000*.

	225x25-01	225x25-08	225x25-10	450x50-13	450x50-16	450x50-18	675x75-23	675x75-25	675x75-29
Best	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(21,0.65)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(21,0.75)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(21,0.85)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(21,0.95)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(31,0.65)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(31,0.75)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(31,0.85)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0
(31,0.95)	20870.0	20511.0	20210.0	76999.0	72540.0	75316.0	183008.0	185892.0	203018.0

Tuning the size of the population

Now, the average required time (in seconds) required by each configuration for solving the instance is displayed. It should be mentioned that the Best row is not necessary in these tables.

Table 5. Results of the average time with $P_{size}=100$.

	225x25-01	225x25-08	225x25-10	450x50-13	450x50-16	450x50-18	675x75-23	675x75-25	675x75-29
(2I,0.65)	2.05	2.08	2.17	7.32	4.01	3.97	9.69	10.70	10.86
(2I,0.75)	2.37	2.34	1.93	7.13	4.35	4.41	11.02	11.41	12.32
(2I,0.85)	2.34	2.32	1.19	8.60	4.12	4.33	10.47	11.21	12.70
(2I,0.95)	2.72	2.84	2.60	9.15	4.89	5.03	11.82	12.97	14.32
(3I,0.65)	3.00	2.94	3.16	11.33	4.90	5.54	12.79	16.05	17.66
(3I,0.75)	3.05	3.11	2.64	11.18	5.61	6.08	13.97	17.36	18.40
(3I,0.85)	3.19	3.30	3.29	13.16	6.05	6.59	14.92	18.00	18.79
(3I,0.95)	4.13	3.93	3.91	12.69	6.74	7.16	19.05	19.96	21.02

Tuning the size of the population

Table 6. Results of the average time with P_{size}=200.

	225x25-01	225x25-08	225x25-10	450x50-13	450x50-16	450x50-18	675x75-23	675x75-25	675x75-29
(2I,0.65)	7.51	7.01	7.24	16.77	11.62	11.92	23.70	22.79	24.19
(2I,0.75)	7.91	6.41	7.90	17.30	12.35	13.36	24.74	23.21	26.06
(2I,0.85)	7.05	7.95	9.12	19.38	13.92	13.39	24.61	26.29	28.15
(2I,0.95)	8.23	8.38	8.24	24.54	13.19	14.64	28.70	28.82	31.88
(3I,0.65)	9.53	10.25	9.67	24.10	16.59	18.19	32.18	34.10	37.20
(3I,0.75)	10.78	10.44	10.02	25.84	18.00	17.43	35.21	35.68	38.86
(3I,0.85)	10.76	10.19	10.05	28.20	20.66	18.42	37.59	39.27	41.97
(3I,0.95)	11.21	11.73	12.07	32.26	18.68	22.52	43.35	42.84	46.67

Tuning the size of the population

Table 7. Results of the average time with $P_{size}=500$.

	225x25-01	225x25-08	225x25-10	450x50-13	450x50-16	450x50-18	675x75-23	675x75-25	675x75-29
(2I,0.65)	19.73	21.15	19.33	45.80	36.72	38.19	68.08	60.21	64.69
(2I,0.75)	21.76	22.12	20.68	50.61	40.85	41.51	74.54	65.63	69.83
(2I,0.85)	23.58	23.70	25.34	55.76	44.50	45.27	81.53	70.96	75.60
(2I,0.95)	28.14	27.61	28.35	67.19	50.75	55.78	91.93	79.05	86.82
(3I,0.65)	29.93	28.48	26.75	67.36	53.84	54.31	102.27	90.62	99.09
(3I,0.75)	30.72	31.60	32.01	76.16	57.73	63.91	110.18	96.98	104.56
(3I,0.85)	37.32	37.52	37.64	83.10	65.77	69.11	117.04	106.54	112.96
(3I,0.95)	42.52	49.77	43.67	94.53	77.50	83.41	145.22	118.26	126.55

Tuning the size of the population

Table 8. Results of the average time with P_{size}=*1000*.

	225x25-01	225x25-08	225x25-10	450x50-13	450x50-16	450x50-18	675x75-23	675x75-25	675x75-29
(2l,0.65)	49.78	60.54	54.48	110.20	86.52	96.67	149.94	130.78	143.95
(2l,0.75)	53.41	65.24	54.40	120.51	95.19	98.76	164.70	142.65	152.40
(2l,0.85)	59.32	70.09	64.15	135.91	110.20	111.43	189.33	157.65	164.06
(2l,0.95)	68.40	86.24	69.13	149.47	128.55	132.47	213.17	175.63	186.12
(3l,0.65)	70.77	79.71	76.31	170.98	128.96	134.82	229.23	212.74	218.01
(3l,0.75)	80.06	91.59	85.40	186.74	145.71	145.43	251.61	218.10	230.04
(3l,0.85)	89.70	102.30	94.86	213.62	159.51	161.56	274.19	243.47	251.94
(3l,0.95)	103.27	116.34	105.87	232.88	190.14	189.81	317.97	273.45	285.42

Tuning the size of the population

It is easy to see from tables 1-4 that the efficient performance of the algorithm is maintained when varying the population size.

In other words, the percent deviations of the average value with respect to the optimum does not show a significant difference.

On the other hand, it is evident from tables 5-8 that the time is increased in a significant way as the population size increased.

Tuning the size of the population

It is easy to see from tables 1-4 that the efficient performance of the algorithm is maintained when varying the population size.

In other words, the percent deviations of the average value with respect to the optimum does not show a significant difference.

On the other hand, it is evident from tables 5-8 that the time is increased in a significant way as the population size increased.

For example, despite the fact that when $P_{size} = 500$ or $P_{size} = 1000$, the algorithm reaches the optimal value in the five runs, the required time is drastically increased.

Tuning the size of the population

It is easy to see from tables 1-4 that the efficient performance of the algorithm is maintained when varying the population size.

In other words, the percent deviations of the average value with respect to the optimum does not show a significant difference.

On the other hand, it is evident from tables 5-8 that the time is increased in a significant way as the population size increased.

For example, despite the fact that when $P_{size} = 500$ or $P_{size} = 1000$, the algorithm reaches the optimal value in the five runs, the required time is drastically increased.

Let's focus when $P_{size} = 100$ or $P_{size} = 200$.

Tuning the size of the population

Table 9. Percentage deviations between the optimal and the algorithm with *tam*=*100*.

	225x25-01	225x25-08	225x25-10	450x50-13	450x50-16	450x50-18	675x75-23	675x75-25	675x75-29
(2I,0.65)	0.00	0.00	0.00	0.05	0.36	1.16	0.64	0.38	0.84
(2I,0.75)	0.00	0.00	0.00	0.04	0.33	0.31	1.06	0.17	0.56
(2I,0.85)	0.00	0.00	0.00	0.15	0.00	0.43	0.05	0.30	0.31
(2I,0.95)	0.00	0.00	0.00	0.04	0.24	0.40	0.07	0.12	0.53
(3I,0.65)	0.00	0.00	0.00	0.05	0.24	0.42	0.59	0.61	0.88
(3I,0.75)	0.00	0.00	0.00	0.03	0.45	0.55	0.60	0.13	0.84
(3I,0.85)	0.00	0.00	0.00	0.05	0.24	0.33	0.32	0.48	0.69
(3I,0.95)	0.00	0.00	0.00	0.39	0.00	0.00	0.18	0.12	0.66

Tuning the size of the population

Table 10. Percentage deviations between the optimal and the algorithm with *tam=200*.

	225x25-01	225x25-08	225x25-10	450x50-13	450x50-16	450x50-18	675x75-23	675x75-25	675x75-29
(2I,0.65)	0.00	0.00	0.00	0.00	0.00	0.09	0.03	0.07	0.32
(2I,0.75)	0.00	0.00	0.00	0.04	0.00	0.00	0.18	0.00	0.43
(2I,0.85)	0.00	0.00	0.00	0.03	0.00	0.00	0.17	0.00	0.18
(2I,0.95)	0.00	0.00	0.00	0.01	0.00	0.00	0.14	0.00	0.25
(3I,0.65)	0.00	0.00	0.00	0.01	0.00	0.00	0.12	0.05	0.32
(3I,0.75)	0.00	0.00	0.00	0.02	0.08	0.00	0.08	0.21	0.00
(3I,0.85)	0.00	0.00	0.00	0.02	0.00	0.00	0.12	0.15	0.12
(3I,0.95)	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.18

Tuning the size of the population

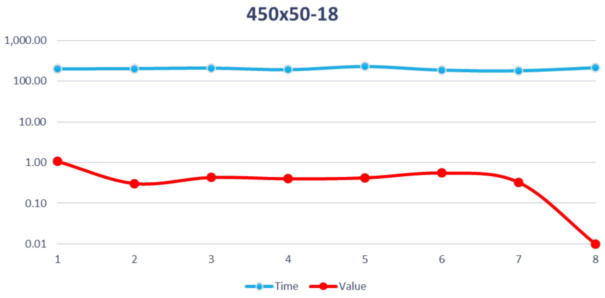
Next figure shows the impact of increasing the population size to 200. The obtained percent deviations for each parameter configuration of a particular instance are plotted.

The blue points represent the deterioration in the required time after increased the population size. The red points indicate the percent improvement in the quality of the leader's objective function by increasing the same parameter.

The two instances with higher difference in the percent deviation were selected. The first figure corresponds to the instance 450x50-18 and the second one to instance 675x75-29. It is worthy to note that a logarithmic scale was used for improving its appreciation.

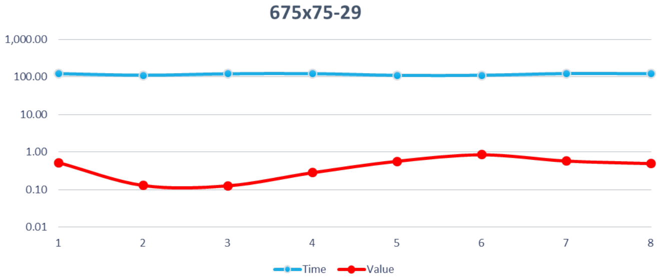
Tuning the size of the population

Figura: Impact from increasing the population size.



Tuning the size of the population

Figura: Impact from increasing the population size.



Tuning parameters *Gen* and *P*

In order to tune both parameters, we must consider only tables 1 and 5.

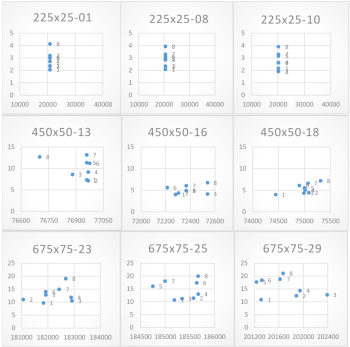
We took into consideration two criteria: the quality of the solution and the required time.

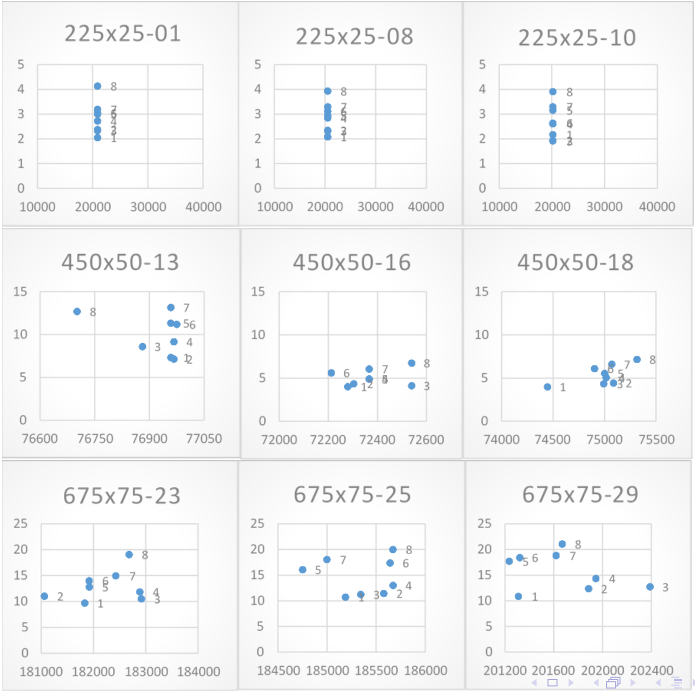
The alternatives for decision will be the 8 possible configurations; for each of them, two corresponding values are associated, one for the quality of the objective function and other for the time.

For making a initial comparison among the tested configurations, the pairs of values (*objective function*, *time*) were plotted for each of the sampled instance.

Tuning parameters *Gen* and *P*

Figura: Illustration of dominance among configurations for each instance.




225x25-01
225x25-08
225x25-10
0 1 2 3 4 5
10000 20000 30000 40000
450x50-13
0 5 10 15
76600 76750 76900 77050
450x50-16
72000 72200 72400 72600
450x50-18
74000 74500 75000 75500
675x75-23
0 5 10 15 20 25
181000 182000 183000 184000
675x75-25
184500 185000 185500 186000
675x75-29
201200 201600 202000 202400

Tuning parameters *Gen* and *P*

It can be appreciate that there are some configurations that are non-dominated in most of the instances. We are going to look for the parameter configuration which dominates the majority of others configurations.

Table 9. Dominance between the configurations.

	(2I,0.65)	(2I,0.75)	(2I,0.85)	(2I,0.95)	(3I,0.65)	(3I,0.75)	(3I,0.85)	(3I,0.95)
225x25-01	8	5	6	4	3	2	1	0
225x25-08	7	5	6	4	3	2	1	0
225x25-10	5	6	7	4	2	3	1	0
450x50-13	4	6	1	3	2	3	0	0
450x50-16	1	1	6	3	2	0	0	0
450x50-18	0	4	1	2	1	0	0	0
675x75-23	1	0	6	4	1	0	0	0
675x75-25	2	5	2	4	0	1	0	0
675x75-29	1	4	5	4	0	0	0	0
Total	29	36	40	32	14	11	3	0

Parameter's selection

The selected values for the parameters are:

$P_{size} = 100$

$Gen = |2I|$

$P = 0{,}85$

Computational environment

A personal computer having a Interl (R) Core processor with 8GB of RAM was used for conducting the computational experimentation.

The code was implemented in Visual Studio 2010 in C++ language.

Two schemes were considered:
(i) The second reformulation presented were solved with CPLEX 12.6.1.
(ii) The genetic algorithm was tested.

Since it is the first time that the MCLBP is studied, no data for comparison exists.

Results for the instances in block M

	REF	GA			
Instance	**Time**	**Best**	**Average**	**Worst**	**Time**
225x25-01	1.09	0.00	0.00	0.00	2.34
225x25-02	1.19	0.00	0.00	0.00	2.36
225x25-03	1.42	0.00	0.00	0.00	2.41
225x25-04	1.05	0.00	0.00	0.00	2.43
225x25-05	1.30	0.00	0.00	0.00	2.26
225x25-06	1.42	0.00	0.00	0.00	2.43
225x25-07	1.70	0.00	0.46	1.15	2.49
225x25-08	1.28	0.00	0.00	0.00	2.32
225x25-09	1.31	0.00	0.00	0.00	2.31
225x25-10	1.75	0.00	0.00	0.00	1.91
450x50-11	10.92	0.00	0.24	1.19	9.06
450x50-12	16.64	0.00	0.37	0.93	8.94
450x50-13	11.33	0.05	0.15	0.56	8.60
450x50-14	8.45	0.00	0.24	0.83	8.74
450x50-15	14.52	0.00	0.26	0.98	8.08
450x50-16	13.75	0.00	0.00	0.00	4.12
450x50-17	9.41	0.00	0.27	0.87	4.40
450x50-18	14.77	0.00	0.43	0.82	4.33
450x50-19	7.52	0.00	0.26	1.30	4.63
450x50-20	5.98	0.00	0.00	0.00	4.29
675x75-21	44.98	0.31	0.87	1.54	10.37
675x75-22	296.02	0.47	0.74	1.35	10.27
675x75-23	25.81	0.00	0.05	0.13	10.47
675x75-24	33.97	0.37	0.65	0.93	10.62
675x75-25	37.44	0.00	0.30	0.66	11.21
675x75-26	15.72	0.00	0.40	1.46	11.73
675x75-27	43.64	0.17	0.31	0.45	12.04
675x75-28	28.42	0.54	0.71	0.97	11.77
675x75-29	83.52	0.00	0.31	0.92	12.70
675x75-30	37.39	0.65	1.01	1.40	12.14

Results for the instances in block L

	REF	GA			
Instance	**Time**	**Best**	**Average**	**Worst**	**Time**
900x100-31	241.75	0.00	0.21	0.41	26.96
900x100-32	466.06	0.00	0.35	0.46	29.70
900x100-33	204.33	0.00	0.23	0.43	28.29
900x100-34	103.66	0.01	0.47	0.82	26.98
900x100-35	1342.30	0.02	0.24	0.39	28.85
900x100-36	176.20	0.10	0.37	0.60	27.77
900x100-37	3270.64	0.00	0.60	0.83	26.89
900x100-38	88.14	0.00	0.20	0.32	28.50
900x100-39	351.01	0.02	0.22	0.30	28.07
900x100-40	182.02	0.10	0.25	0.39	26.43
1350x150-41	10800.00	0.10	0.39	0.66	85.48
1350x150-42	10800.00	0.68	0.89	1.23	93.07
1350x150-43	10800.00	0.36	0.55	0.65	91.63
1350x150-44	10800.00	-0.45	-0.22	0.02	84.02
1350x150-45	10800.00	0.45	0.73	0.94	80.41
1350x150-46	309.64	0.35	1.51	2.02	83.38
1350x150-47	10800.00	0.05	0.60	0.87	83.63
1350x150-48	10800.00	2.29	2.48	2.60	82.46
1350x150-49	3952.89	0.41	0.63	0.81	84.70
1350x150-50	10800.00	0.76	1.38	1.77	89.86
1800x200-51	10800.00	0.90	1.18	1.48	231.45
1800x200-52	10800.00	0.64	0.91	1.05	251.10
1800x200-53	10800.00	-0.16	0.44	0.62	204.46
1800x200-54	10800.00	0.36	0.70	0.97	221.57
1800x200-55	10800.00	-0.30	-0.14	0.11	223.89
1800x200-56	10800.00	0.57	1.07	1.53	223.89
1800x200-57	10800.00	0.11	0.34	0.60	210.52
1800x200-58	10800.00	-0.50	-0.06	0.21	216.30
1800x200-59	10800.00	-0.03	0.27	0.58	228.00
1800x200-60	10800.00	0.76	1.16	1.44	236.82

Conclusions

In this paper, the maximal covering location bi-level problem was introduced.

In order to solve the problem, a classic technique for reducing the bi-level problem into a single-level one was applied. Two schemes for assuring optimality were considered.

Also, linearization techniques were considered in order to have a mixed-integer programming problem. A commercial optimizer was used for solving the MCLBP showing inefficiency for large size instances.

Conclusions

Therefore, a genetic algorithm was developed for reducing the computational time without affecting in a significant manner the quality of the obtained solutions.

In many of the instances the GA reaches optimality in less time than CPLEX; in the cases where the optimal value was not found, the optimality gap is less than 1 % (excepts from one instance).

Moreover, in some instances the GA obtained the best objective function value known.

Current research

In order to solve large scale instances motivated by big fast food chains, a hybrid algorithm which considers GRASP and tabu search being developed.

Also, an alternative single-level formulation that models the same problem could be modelled. An the corresponding equivalence is being demonstrated.

An interesting constraint that could be included in the model that is motivated for keeping closed facilities which does not meet a minimum amount of captured demand. *(coupling constraints)*

Camacho-Vallejo, J.F., Cordero-Franco, A.E., González-Ramírez, R.G., 2014. Solving the bilevel facility location problem under preferences by a Stackelberg-evolutionary algorithm. *Mathematical Problems in Engineering*, vol. 2014, 14 pages.

Hansen, P., Kochetov, Y., Mladenovic, N., 2004. Lower bounds for the uncapacitated facility location problem with user preferences. *Preprint G-2004-24, Mart 2004, GERAD-HEC, Montreal, Canada*.

Lee, J.M., Lee, Y.H., 2012. Facility location and scale decision problem with customer preference. *Computers and Industrial Engineering*, vol. 63, pp. 184-191.

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*, vol. 39, pp. 43-50.

Resende, M., 1998. Computing approximate solutions of the maximum covering problem with GRASP. *Journal of Heuristics*, vol. 4, pp. 161-177.

Vasil'ev, 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*, vol. 4(3), pp. 441-454.

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*, vol. 49(6), pp. 1010-1020.

IWOBIP 2016

Dates: March, 7th - 11th, 2016, in the FCFM, UANL, Mexico

web page:
http://bi-level.fcfm.uanl.mx

IWOBIP 2016

Confirmed Plenary Speakers

Martine Labbé (Université Libre de Bruxelles, Belgium)
Boris Mordukhovich (Wayne State University, USA)
Vyacheslav Kalashnikov (Tecnológico de Monterrey, Mexico)
Leo Liberti (École Polytechnique, France)

IWOBIP 2016

Confirmed Plenary Speakers

Martine Labbé (Université Libre de Bruxelles, Belgium)
Boris Mordukhovich (Wayne State University, USA)
Vyacheslav Kalashnikov (Tecnológico de Monterrey, Mexico)
Leo Liberti (École Polytechnique, France)

Confirmed Mini-courses

Stephan Dempe (TU Bergakademie Freiberg, Germany)
Ankur Sinha (Indian Institute of Management Ahmedabad, India)
Jakub Mareček (IBM Research, Ireland)

IWOBIP 2016

Confirmed Plenary Speakers
Martine Labbé (Université Libre de Bruxelles, Belgium)
Boris Mordukhovich (Wayne State University, USA)
Vyacheslav Kalashnikov (Tecnológico de Monterrey, Mexico)
Leo Liberti (École Polytechnique, France)

Confirmed Mini-courses
Stephan Dempe (TU Bergakademie Freiberg, Germany)
Ankur Sinha (Indian Institute of Management Ahmedabad, India)
Jakub Mareček (IBM Research, Ireland)

There are scholarships for students (accommodation, meals and registration fee).