# Using the Constructive Genetic Algorithm for Solving the Probabilistic Maximal Covering Location-Allocation Problem

Francisco de Assis Corrêa
*Instituto Nacional de Pesquisas Espaciais*
*São José dos Campos - SP, Brasil*
*fcorrea@directnet.com.br*

Luiz Antonio Nogueira Lorena
*Instituto Nacional de Pesquisas Espaciais*
*São José dos Campos – SP, Brasil*
*lorena@lac.inpe.br*

## Abstract

*The Maximal Covering Location Problem (MCLP) maximizes the population that has a facility within a maximum travel distance or time. Numerous extensions have been proposed to enhance its applicability, like the probabilistic model for the maximum covering location-allocation with constraint in waiting time or queue length for congested systems, with one or more servers per service center. In this paper we present one solution procedure for that probabilistic model, considering one server per center, using the Constructive Genetic Algorithm. The results of tests on the solution procedure are presented.*

## 1. Introduction

The Maximal Covering Location Problem (MCLP) has been extensively studied in the literature since its formularization made by Church and ReVelle (1974). The main objective of the MCLP is to choose the location of facilities to maximize the population that has a facility within a maximum travel distance (or time). Thus, a population is considered covered if it is within a predefined service distance (or time) from at least one of existing facilities. Considerable revision of this subject can be found in Hale and Moberg (2003), Serra and Marianov (2004) and Galvão (2004). The MCLP does not require that all demand areas be covered, but offers attendance to the maximum population, considering the available resources. Some useful applications are extensions of this formularization.

In many works involving location problems, the distance (or time) between demand points and the facilities to which they are being located are the factor that represents the quality of the services that are given to the users. However, when service networks are projected, as health systems or banking, the location of service centers has a strong influence on the congestion of each of them, and, consequently, the quality of services must be better defined and not only considering the travel distance or time. The centers must be located to allow the users to arrive at the center in an acceptable time, is also desirable that the waiting time for attendance is no longer than a given time limit or nobody stands on line with more than a predetermined number of other clients. These are important parameters in the measure of the desired quality [11].

Congestion happens when a center is not capable to deal, simultaneously, to all the service requests that are made to it. Normally, the traditional models that deal with congestion include a capacity constraint, which forces the demand for service, normally constant in time and equal to an average, to be smaller than the maximum capacity of the center all the time. This is a deterministic approach to the problem, because does not consider the dynamic nature of the congestion. Depending on how the capacity constraint is developed, this makes that the solution model presents idle servers, or is a system that is not capable to attend all the demand [11][12].

Marianov and Serra (1998) proposed models based on the fact that the number of requests for services are not constant in time, but a stochastic process, whose stochasticity of demand is explicitly considered in the capacity constraints. Instead of being limited to a maximum value, the authors define a minimum limit for the quality of the service reflected in the waiting time or the number of people waiting for service.

Those researchers address the formulation of several maximal coverage models, with one or more servers per service center, so that all the population is served within a standard distance (or time), and nobody stands on line more than a given time limit, or with more than a given number of other clients, with a probability of at least $\varphi$.

The purpose of this paper is to examine the Queuing Maximal Covering Location-Allocation Model (QM-CLAM) with one server per service center, proposed by Marianov and Serra (1998), and present a solution using the Constructive Genetic Algorithm (CGA). The QM-CLAM is briefly discussed in Section 2, the CGA is described in Section 3, computational results are reported in Section 4 and conclusions are presented in Section 5.

## 2. QM-CLAM

The traditional Maximum Covering Location Problem (MCLP) proposed by Church and ReVelle (1974) can not be used to deal with the congestion constraints, because there are no allocation variables. Then, it is impossible to compute the requests of services that arrive at a center, and, consequently, to determine when congestion occurs.

Thus, the MCLP has been rewritten as p-median-like model, modified to accommodate the location and allocation variables. The objective is to maximize the covered population, considering a predefined number of service centers.

An integer linear programming formulation for the QM_CLAM is obtained by introducing the following variables. Let $y_j = 1$ if a center is located at a node $j$ and $y_j = 0$ otherwise; $x_{ij} = 1$ if the users located at demand node $i$ is allocated to a center located at $j$, and $x_{ij} = 0$ otherwise. We consider $i \in I$ and $j \in N_i$, such as $I$ is a set of demand nodes, and $N_i$ is either a set of candidate locations that are within a standard distance from node $i$, or the set of candidate locations which can be reached from node $i$, within a certain standard time. Let $a_i$ be the total population at demand node $i$. The QM_CLAM can be formally stated as:

$$\text{v(QM-CLAM)} = \text{Max} \sum_{i,j} a_i x_{ij} \qquad (1)$$

Subject to

$$x_{ij} \le y_j \qquad \forall i, j \qquad (2)$$

$$\sum_{j \in N_i} x_{ij} \le 1 \qquad \forall i \qquad (3)$$

$$\sum_{i \in I} z_i x_{ij} \le \mu_j \sqrt[b+2]{1-\varphi} \qquad \forall j \qquad (4)$$

$$\text{or } \sum_{i \in I} z_i x_{ij} \le \mu_j + \frac{1}{\tau} \ln(1-\varphi) \quad \forall j \qquad (4a)$$

$$\sum_i y_i = p \qquad (5)$$

$$y_j, x_{ij} \in \{0,1\}, \quad j \in N_i \qquad \forall i, j \qquad (6)$$

The objective (1) maximizes the population allocated to a center. Constraint (2) defines that a demand point $i$ can be allocated to a node $j$ only if there is a center in $j$. Constraint (3) forces each demand node $i$ to be allocated to at most one service center $j$. Constraints (4) force to each center has no more than $b$ people on a line, with a probability of at least $\varphi$. Constraints (4a), make the total time spent by a user at a center $j$ shorter than or equal to $\tau$, with a probability of at least $\varphi$. Constraint (5) sets the number of centers to be located. Constraints (6) define the integrality requirements.

In order to write constraint (4), the authors made assumption that requests for service at each demand node $i$ appear according to a Poisson process with intensity $z_i$. The service requests at a center are the union of the requests for service of the demand nodes, and they can be described as another stochastic process, equal to the sum of several Poisson processes, with intensity $\lambda_j$:

$$\lambda_j = \sum_{i \in I} z_i x_{ij} \qquad (7)$$

Which means that, if the variable $x_{ij}$ is one, node $i$ is allocated to center $j$ and the corresponding intensity $z_i$ will be included in computation of $\lambda_j$. They have considered the well known results for a M/M/1 queuing system for each center and its allocated users [8]. An exponentially distributed service time, with an average rate $\mu_j$, has been considered in those models, where $\mu_j \ge \lambda_j$, otherwise the system does not reach the equilibrium.

The QM-CLAM belongs to the NP-Complete class of problems [16]. Even using commercial solvers is not always possible to find the optimal solution in a reasonable computational time, due to its classification and to the problem size. Therefore, alternative methods are investigated. In the next chapter a solution based on the Constructive Genetic Algorithm will be presented.

## 3. Constructive Genetic Algorithm

Genetic algorithms have been developed by John Holland, his colleagues, and his students at the University of Michigan. Refinements of the method have been implemented in the following decades. Some of them and basic information can be achieved in Lacerda and Carvalho (1999) and Goldberg (1989).

The Constructive Genetic Algorithm (CGA) [2][9][13][14] works with a population formed by *schemata* (incomplete solutions) and *structures* (complete solution).

In this work, structures and schemata can be generated by a string representation, using the symbols **1**, **2** and #. The schemata make an explicit reference to the symbol # ("do not care") and represent a population of partial solutions, who is a base for the construction of a population with complete better solutions, throughout the evolutionary process. The symbol **1** represents a center; the symbol **2** represents a demand point to be allocated to a center and the "do not care" symbol represents a non defined point to the problem, which will become a center or a demand point during the evolutionary process.

The Figure 1 shows an example of an individual S for a problem with 10 demand points and 2 centers. Constraints (2), (3) and (5) of QM_CLAM are implicitly considered in this representation. The capacity constraint (4) or (4a) is considered during a demand point allocation, when adding its intensity $z_i$, the remaining capacity of the center is not exceeded. QM_CLAM is a clustering problem, and any clustering algorithm will attempt to determine some inherent or natural grouping in the data, using distance or similarity measures between individual data [9]. In this case, once a center is chosen, a cluster is determined by allocating demand points to it that are within a coverage distance, that satisfies its capacity and are not allocated to any other center.

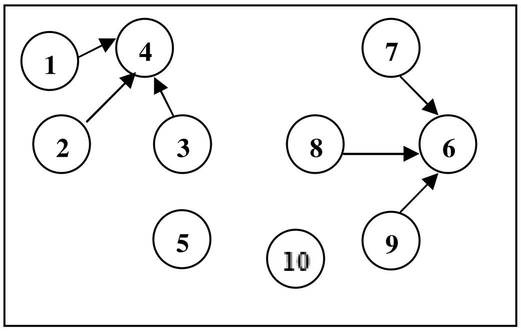

**Figure 1. S = (2221#1222#) (Furtado (1998)).**

The CGA has the traditional operators: selection, recombination and mutation, and differs from a classical GA in the way to evaluate the schemata (*fg-fitness*), in the possibility to use heuristics to define the fitness evaluation function and in the treatment of a dynamic population [14].

The *fg-finess* is a double fitness evaluation of an individual $S_k \in P_\alpha$, where $P_\alpha$ is a population at evolutionary instant $\alpha$. The $f$ value represents the objective function, $f$: $P_\alpha \rightarrow R^+$, expression (1), and the g value is calculated by a heuristic, $g$: $P_\alpha \rightarrow R^+$,such that $g(S_k) \geq f(S_k)$, to all $S_k \in P_\alpha$. The first evaluates the quality of individual and the second applies a problem-specific heuristic (called training heuristic) to evaluate the neighborhood of individual, being the value of the best solution found attributed to $g$.

In this work, the heuristic used for $g$ calculation is based on the algorithm used to improve the primal solutions in Pereira and Lorena (2001). This heuristic searches for a new center in each cluster, swapping the current center with a non-center node in the same cluster, changing the allocation solution. This change may alter both allocation and covering configuration, so an algorithm for recalculating the coverage is needed and was implemented. The algorithm stops when swapping the current center do not improve the value of the best reallocation.

The Constructive Genetic Algorithm works with a dynamic population, initially formed only by *schemata*, which is enlarged after the use of recombination operators, or made smaller along the generations, guided by an evolutionary parameter. That population is built, generation after generation, by directly searching for well-adapted *structures* (complete solution) and also for good schemata.

The evolutionary process considers an adaptive rejection threshold, which defines the adaptation of an individual. This adaptation is proportional to its *ranking* $\delta$, calculated by equation (8),

$$\delta = \frac{d.G_{\max} - [g(S_k) - f(S_k)]}{d.[G_{\max} - g(S_k)]} \qquad (8)$$

that is composed by:

- A component concerning the adaptation of individual in relation to the training heuristic (*g-f*).
- A component ($G_{\max} - g(S_k)$) that privileges the maximization of the function g, calculating the distance between the individual and an upper bound for all possible values for the $f$ and $g$ functions ($G_{\max}$).
- A constant $d$, $0 \leq d \leq 1$, to balance the components of the equation.

Thus, better individuals have greater ranks. The initial population $P_0$ is made by schemata and, when the individuals receive their correspondents ranking values. The individuals are sorted by decreasing values of $\delta$. The population is then controlled in a dynamic way (see Figure 2) by an adaptative rejection threshold $\alpha$, calculated by equation (9), that uses the current population size $|P|$, the best ($\delta_1$) and the worst ($\delta_{|P|}$) rankings of individuals in current population, the estimated remaining number of generations, RG, the constant $\varepsilon$ that controls the speed of evolutionary process, and *l* that guarantee a minimum step in that process.

$$\alpha_t = \alpha_{t-1} + \varepsilon . | P | . \frac{(\delta_1 - \delta_{|P|})}{RG} + l \qquad (9)$$

The initial α value is the worst ranking of individuals ($\delta_{|P|}$) in the initial population. At the end of each generation the individulas less adapted ($\delta(S_k) \leq \alpha$) are eliminated from the population. The best individual for each generation is kept to define, at the end of the evolutionary process, the best solution found.

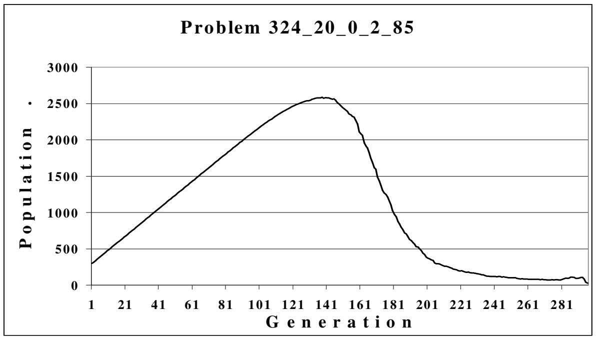

**Figure 2. Population at each generation**

Two structures or schemata are selected for recombination The first, called *base*, is obtained from the 20% better individuals of the population ($S_{base}$). The second, called *guide*, is randomly selected out from the whole population ($S_{guide}$). In the recombination operation, the current labels in corresponding positions are compared. Let $S_{new}$ be the new offspring after recombination. The structure or schema $S_{new}$ is obtained by applying the folowing operations, based on Furtado (1998) and Oliveira and Lorena (2005):

- $S_{base}$ = # and $S_{guide}$ = # then $S_{new}$ = #
- $S_{base}$ = 1 and $S_{guide}$ = 1 then $S_{new}$ = 1
- $S_{base}$ = 2 and $S_{guide}$ = 2 then $S_{new}$ = 2
- $S_{base}$ = 1 and $S_{guide}$ = # then $S_{new}$ = 1
- $S_{base}$ = 2 and $S_{guide}$ = # then $S_{new}$ = 2
- $S_{base}$ = # and $S_{guide}$ = 2 then $S_{new}$ = 2
- $S_{base}$ = # or 2 and $S_{guide}$ = 1 then $S_{new}$ = 1 or 2, chosen randomly
- $S_{base}$ = 1 and $S_{guide}$ = 2 then $S_{new}$ = 1 ou 2, chosen randomly

After a recombination operation, more than *p* values 1 can appear in the $S_{new}$. That condition is not relaxed and $S_{new}$ must be validated to guarantee the constraint (5), by inserting or removing values 1 from that offspring.

The mutation operator changes a center with a neighbor that it is not covered by any center. This increases the possibility of all the demand points to be centers.

## 4. Computational results

The CGA was tested in the 30-node network provided by Marianov and Serra (1998) and in a 324-node network gotten from a geographical data base of São José dos Campos-SP, Brasil, increased by fictitious population in each demand point. The last one is available at http://www.lac.inpe.br/~lorena/instancias.html.

By varying the p, b, μ, φ e τ parameters, various problems have been created. The results from CGA have been compared to the results obtained using the commercial solver CPLEX, version 7.5 [7].

For the implementation of the QL_CLAM, the service centers are primary health care centers, with one physician at each center. Each demand point is also a potential center location, and the distances are Euclidean. To the 30-node network, it has been considered: covered distance equals to 1,5 miles; average service time (1/μ) was set at 20 minutes; call rates were set at 0.015 times the node population for the constrained queue length and 0,006 times the node population for the constrained waiting time, all defined on Marianov and Serra (1998). To the 324-node network, it has been considered: covered distance equals 250 meters, average service time (1/μ) was set at 15 minutes; call rates were set at 0,01 times the node population for both constrained queue length and constrained waiting time.

The problems have been codified in the following way: number of points, number of center, constraint type (0 for the constrained queue length and 1 for the constrained waiting time), number of people in line or waiting time, and probability. Example: 324_20_0_2_95, which means 324 points, 20 centers, constrained queue length, maximum of two people in line, with the probability, at least, 95%. The CGA code has been written in Object Pascal. The times in tables are shown in seconds and have been determined in a Pentium IV 3 GHz computer, with 1Gb of RAM memory, for the 324-node network, and in a Pentium III 800 MHz computer, with 384 Mb of RAM, for the 30-node network. The times for the attainment of the CPLEX solutions was limited in 2 hours (7200 seconds), except for the instances marked with one *, that it defines a stop in the execution due to out of memory error. The results of the CGA have been gotten using the values of the parameters shown in Table 1.

**Table 1. CGA parameters**

| Parameters | 30 and 324-node network |
|---|---|
| $G_{Max}$ | 1,1 times the sum of the population of all demand points |
| d | 0.1 |
| ε | 0.001 |
| *l* | 0.0001 |

| | |
|---|---|
| Crossover / generation | 30 |
| Mutation probability | 0.20 |
| Initial population | 300 |
| Maximal number of generations | 300 |

The tables 2 and 3 show the results obtained for the 324-node network and tables 4 and 5 show the results for the 30-node network. Tables 2 and 4 show the best integer solutions found and the Gap Cplex to the 324-node and 30-node networks, respectively. The values of Gap Cplex equal zero define that the optimal has been achieved. Tables 3 and 5 show the results found by CGA to the 324-node and 30-node networks, respectively.

**Table 2. CPLEX results for the 324-node network**

| Problem | CPLEX | |
|---|---|---|
| | CPLEX Solution | Gap Cplex (%) |
| 324_10_0_0_95 | 21365 | 0.47 |
| 324_10_0_1_95 | 35050 * | 0.90 |
| 324_10_0_2_95 | 45232 | 0.36 |
| 324_10_0_0_85 | 36802 | 1.03 |
| 324_10_0_1_85 | 50037 | 1.25 |
| 324_10_0_2_85 | 58516 | 2.10 |
| 324_20_0_0_95 | 41676 | 3.01 |
| 324_20_0_1_95 | 68443 | 3.35 |
| 324_20_0_2_95 | 88269 | 2.86 |
| 324_20_0_0_85 | 71119 | 4.56 |
| 324_20_0_1_85 | 95917 | 6.36 |
| 324_20_0_2_85 | 112691* | 6.03 |
| 324_10_1_40_85 | 27307 | 1.45 |
| 324_10_1_41_85 | 28710 * | 2.30 |
| 324_10_1_42_85 | 30735 | 0.72 |
| 324_10_1_48_90 | 26360 | 2.13 |
| 324_10_1_49_90 | 28249 | 0.29 |
| 324_10_1_50_90 | 29338 | 1.19 |
| 324_20_1_40_85 | 53198 * | 4.15 |
| 324_20_1_41_85 | 55740 * | 5.38 |
| 324_20_1_42_85 | 60783 * | 1.86 |
| 324_20_1_48_90 | 52617 | 2.33 |
| 324_20_1_49_90 | 54097 * | 4.75 |
| 324_20_1_50_90 | 57974 | 2.45 |

The results for the CGA reflect fifty executions of each problem and are shown in four columns: the best value found (column Best solution), the average value (column Average solution), the average time (column Time) and the column "Deviation", that reflects the relative error of the average solution for the CGA, relative to the best found primal solution, and are calculated by (CPLEX solution – Average solution)/(CPLEX solution). Therefore, the negative values of the deviations indicate that the average solution for the CGA was better than the CPLEX solution. The values in boldface show the best solutions found.

**Table 3. CGA results for the 324-node network**

| Problem | CGA | | | |
|---|---|---|---|---|
| | Best solution | Average solution | Time (s) | Deviation (%) |
| 324_10_0_0_95 | **21431** | 21373 | 8.63 | -0.04 |
| 324_10_0_1_95 | **35342** | 35304 | 7.79 | -0.03 |
| 324_10_0_2_95 | **45347** | 45245 | 7.81 | -0.03 |
| 324_10_0_0_85 | **37145** | 37069 | 8.10 | -0.38 |
| 324_10_0_1_85 | **50880** | 50711 | 7.69 | -0.29 |
| 324_10_0_2_85 | **59624** | 59437 | 7.62 | -0.27 |
| 324_20_0_0_95 | **42577** | 42318 | 24.30 | -0.76 |
| 324_20_0_1_95 | **70471** | 70308 | 23.40 | 0.05 |
| 324_20_0_2_95 | **89970** | 89355 | 24.15 | 0.15 |
| 324_20_0_0_85 | **73407** | 73001 | 23.69 | 0.06 |
| 324_20_0_1_85 | **99576** | 98353 | 24.69 | 1.22 |
| 324_20_0_2_85 | **116639** | 115235 | 23.33 | 1.43 |
| 324_10_1_40_85 | **27675** | 27602 | 8.54 | -0.15 |
| 324_10_1_41_85 | **29324** | 29260 | 8.27 | -0.42 |
| 324_10_1_42_85 | **30932** | 30895 | 8.34 | -0.09 |
| 324_10_1_48_90 | **26883** | 26835 | 8.35 | -0.13 |
| 324_10_1_49_90 | **28280** | 28221 | 8.28 | 0.10 |
| 324_10_1_50_90 | **29641** | 29593 | 8.26 | -0.03 |
| 324_20_1_40_85 | **54804** | 54414 | 23.92 | 0.18 |
| 324_20_1_41_85 | **58009** | 57571 | 24.70 | 0.47 |
| 324_20_1_42_85 | **61545** | 61266 | 23.81 | -0.79 |
| 324_20_1_48_90 | **53300** | 52958 | 24.32 | 0.21 |
| 324_20_1_49_90 | **56216** | 55813 | 24.15 | -0.44 |
| 324_20_1_50_90 | **58941** | 58577 | 26.03 | 0.39 |

**Table 4. CPLEX results for the 30-node network**

| Problem | CPLEX | | |
|---|---|---|---|
| | CPLEX Solution | Gap Cplex (%) | Time (s) |
| 30_2_0_0_85 | 3700 | 0.49 | 7200 |
| 30_3_0_0_85 | 5390 | 0 | 2 |
| 30_2_0_1_85 | **5100** | 0 | 38 |
| 30_3_0_1_85 | 5390 | 0 | 2 |
| 30_2_0_2_85 | 5210 | 0 | 0 |
| 30_3_0_2_85 | 5390 | 0 | 1 |
| 30_5_0_0_95 | 5330 | 0.69 | 7200 |
| 30_6_0_0_95 | 5410 | 0 | 44 |
| 30_3_0_1_95 | **5270** | 0 | 390 |
| 30_4_0_1_95 | 5390 | 0 | 0 |
| 30_2_0_2_95 | 4520 | 0.43 | 7200 |
| 30_3_0_2_95 | 5390 | 0 | 0 |
| 30_4_1_48_90 | 1920 | 1.47 | 7200 |
| 30_5_1_48_90 | **2400** | 1.47 | 7200 |
| 30_3_1_49_90 | 2160 | 0 | 2 |
| 30_4_1_49_90 | 2880 | 0 | 0 |
| 30_5_1_50_90 | 4700 | 0.81 | 7200 |
| 30_6_1_50_90 | 5390 | 0 | 1 |
| 30_5_1_40_85 | **3050** | 1.19 | 7200 |
| 30_6_1_40_85 | 3600 | 2.88 | 7200 |
| 30_7_1_40_85 | 4060 | 0 | 1 |
| 30_6_1_41_85 | **5330** | 0.74 | 7200 |
| 30_7_1_41_85 | **5410** | 0 | 27 |
| 30_8_1_41_85 | 5470 | 0 | 0 |
| 30_4_1_42_85 | 4600 | 0.81 | 7200 |
| 30_5_1_42_85 | 5390 | 0 | 1 |

Considering the 324-node network, the CGA has better results in 100% of the tests, in competitive times related to the CPLEX. For the 30-node network, the CGA supplied, in terms of average values, results equal or better than CPLEX in 58% of the tests, including optimal values.

**Table 5 – CGA results for the 30-node network**

| Problem | CGA | | | |
|---|---|---|---|---|
| | Best solution | Average solution | Time (s) | Deviation (%) |
| 30_2_0_0_85 | 3700 | 3700 | 0.31 | 0.00 |
| 30_3_0_0_85 | 5390 | 5390 | 0.51 | 0.00 |

| | | | | |
|---|---|---|---|---|
| 30_2_0_1_85 | 5090 | 5090 | 0.36 | 0.20 |
| 30_3_0_1_85 | 5390 | 5390 | 0.48 | 0.00 |
| 30_2_0_2_85 | 5210 | 5210 | 0.38 | 0.00 |
| 30_3_0_2_85 | 5390 | 5390 | 0.47 | 0.00 |
| 30_5_0_0_95 | 5330 | 5323 | 0.80 | 0.00 |
| 30_6_0_0_95 | 5410 | 5392 | 0.84 | 0.00 |
| 30_3_0_1_95 | 5240 | 5240 | 0.48 | 0.57 |
| 30_4_0_1_95 | 5390 | 5390 | 0.54 | 0.00 |
| 30_2_0_2_95 | 4520 | 4513 | 0.30 | 0.15 |
| 30_3_0_2_95 | 5390 | 5390 | 0.50 | 0.00 |
| 30_4_1_48_90 | 1920 | 1920 | 0.66 | 0.00 |
| 30_5_1_48_90 | 2390 | 2390 | 0.74 | 0.42 |
| 30_3_1_49_90 | 2160 | 2160 | 0.53 | 0.00 |
| 30_4_1_49_90 | 2880 | 2877 | 0.59 | 0.10 |
| 30_5_1_50_90 | 4700 | 4700 | 0.73 | 0.00 |
| 30_6_1_50_90 | 5390 | 5390 | 0.97 | 0.00 |
| 30_5_1_40_85 | 3020 | 3001 | 0.74 | 1.61 |
| 30_6_1_40_85 | **3610** | **3610** | 0.91 | -0.28 |
| 30_7_1_40_85 | 4060 | 4060 | 1.03 | 0.00 |
| 30_6_1_41_85 | 5300 | 5274 | 1.00 | 1.05 |
| 30_7_1_41_85 | 5390 | 5390 | 0.88 | 0.37 |
| 30_8_1_41_85 | 5470 | 5470 | 0.95 | 0.00 |
| 30_4_1_42_85 | 4600 | 4600 | 0.61 | 0.00 |
| 30_5_1_42_85 | 5390 | 5390 | 0.77 | 0.00 |

## 5. Conclusions

This work presented a solution for the probabilistic maximal covering location-allocation problem using CGA. The results show that the CGA approach is competitive for the resolution of this problem in reasonable computational times. For some instances of 30-node network, the optimal values have been found. Therefore, these results validate the CGA application to the QM-CLAM.

## 6. References

[1] R.L. Church and C. ReVelle, Maximal covering location problem. *Papers of the Regional Science Association*, 1974, 32: 101-118.

[2] J.C. Furtado, Algoritmos genéticos construtivos na otimização de problemas combinatoriais de agrupamentos. 1998. 112 p. Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 1998.

[3] R.D. Galvão, Uncapacitated facility location problems: contributions. *Pesquisa Operacional*, 2004, 24: 7-38.

[4] D.E. Goldberg, *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, Reading, MA, 1989.

[5] T.S. Hale and C.R. Moberg, Location science review. *Annals of Operations Research*, 2003, 123: 21-35.

[6] ILOG CPLEX 7.5 Reference Manual 7.5v. 610p. Copyright by ILOG, France, 2001.

[7] E.G.M Lacerda and A.C.P.L.F de Carvalho, Introdução aos algorítmos genéticos. In: XIX Congresso Nacional da Sociedade Brasileira de Computação, 19, 1999. Rio de Janeiro. **Anais**… Rio de Janeiro: EntreLugar, 1999, v. 2, p. 51-126.

[8] R.C. Larson and A. R. Odoni, *Urban operations research*, Prentice Hall, Englewood Cliffs, N.J:, 1981.

[9] L.A.N. Lorena and J.C. Furtado, Constructive genetic algorithm for clustering problems. *Evolutionary Computation*, 2001, 9(3): 309-327.

[10] L.A.N. Lorena and M.A. Pereira. A lagrangean/surrogate heuristic for the maximal covering location problem using Hillsman´s edition, *International Journal of Industrial Engineering*, 2002, 9: 57-67.

[11] V. Marianov and D. Serra. Probabilistic maximal covering location-allocation models for congested systems. *Journal of Regional Science*, 1998, 38(3): 401-424.

[12] V. Marianov and D. Serra. Hierarchical location-allocation models for congested systems. *European Journal of Operational Research*, 2001, 135: 195-208.

[13] A.C.M. Oliveira and L.A.N. Lorena, Detecting promising areas by evolutionary clustering search. *Advances in Artificial Intelligence Series*, 2004, 385-394.

[14] A.C.M. Oliveira and L.A.N. Lorena, Population training heuristics. *Lecture Notes in Computer Science*, 2005, 3448: 166–176.

[15] M.A. Pereira and L.A.N. Lorena, A heurística Lagrangeana/Surrogate aplicada ao problema de localização de máxima cobertura. In: XXXIII Simpósio Brasileiro de Pesquisa Operacional (SBPO), 2001, Campos do Jordão. **Anais**… p. 1326-1337.

[16] H. Pirkul and D.A. Schilling, The maximal covering location problem with capacities on total workload. *Management Science*, 1991, 37(2): 233-248.

[17] D. Serra and V. Marianov, New trends in public facility location modeling. Universitat Pompeu Fabra Economics and Business Working Paper 755, 2004. Available at <http://www.econ.upf.edu/docs/papers/downloads/755.pdf>.