A Constructive Genetic Algorithm for the Maximal Covering Location Problem

*Reinaldo Gen Ichiro Arakaki** *Luiz Antonio Nogueira Lorena**

* Brazilian Institute of Space Research (INPE)
C.P. 515 - CEP 12201-970 - São José dos Campos - SP - Brazil
Email: rgia@nucleo.inpe.br, lorena@lac.inpe.br

1 Introduction

Genetic Algorithms (GAs) have been considered in recent years as powerful tools to solve *optimization* problems ([7]; [8]). The underlying foundations for such algorithms are the controlled evolution of a structured population. Holland [5] outlined *GAs* and their mechanisms (schemata formation and propagation over generations).

The *Constructive Genetic Algorithm (CGA)* was proposed by Lorena and Furtado [6], as an alternative to the traditional *GAs* approach. Particularly the *CGA* directly evaluates schemata, works with a dynamic population size, performs mutation in complete structures, and uses heuristics in schemata and/or structure representation. The schemata are evaluated and can be added to the population if they satisfy an evolution test. Structures can result from recombination of schemata or complementing of good schemata. A mutation process is applied just to structures and the best structure generated is kept in the process.

In this paper, we apply the *CGA* to the maximal covering location problem (*MCLP*)[3][4]. The *MCLP* is the problem of locating p facilities on a network such that the maximal population is attended (or covered) within a given service distance S ([2]). The *MCLP* has applications in the public and private sectors. Chung [1] and Schilling et al. [9] review several applications of *MCLP*.

We report computational tests for real world data ranging from 324 to 500 vertices using georeferenced data of São José dos Campos city, Brazil and random generated scenarios for 100 and 150 vertices.

2 CGA modeling

In this section is described the *CGA* approach. Two fitness functions are defined on the space of all schemata and structures representing the location problem. The evolution process considers the two objectives on an adaptive rejection threshold, which gives ranks to individuals in population and yields a dynamic population.

2.1 Structure and schema representations

The structure and schema representations for the *MCLP* use a binary alphabet.

Suppose a given graph *G=(V,E)*. A typical instance of the problem is composed of n demand points (vertices) $V = \{1,...,n\}$, demands D_i, a service distance S, and a distance (weight) matrix $[\mu_{ji}]$, indicating distances between pairs of vertices, such that $\mu_{ji} \geq 0$, $\mu_{jj} = 0$ and $\mu_{ji} = \mu_{ij}$ for all $j, i \in V$.

To define the representation p vertices are elected as the *seeds* (located facilities), i. e., the initial vertices in clusters that will cover the other vertices that participate in the representation and are situated within the range S.

The representation of structures and schemata will be:

s_k = (0,0,1,1,#,0,0,#,#,1), where

the numbers 1 indicate the seeds and their positions in the string (representing vertices). The number 0 indicates that the corresponding vertex may be covered by a seed and # that the vertex was considered temporarily out of the problem. A structure representation has no #s. For each schema or structure s_k, exactly p clusters $C_1(s_k), C_2(s_k), ..., C_p(s_k)$ are identified. If a vertex is within the range of two or more seeds it is included on these clusters.

2.2 The bi-objective optimization problem

Let X be the set of all structures and schemata that can be generated by $0-1-\#$ string representation of the later section, and consider a double fitness evaluation (*fg-fitness*) f and g, defined as $f : X \to \Re_+$ and $g : X \to \Re_+$ such $f(s_k) \leq g(s_k)$, for all $s_k \in X$.

The *CGA* optimization problem implements the *fg-fitness* with the following two objectives:

(*interval minimization*) Search for $s_k \in X$ of minimal $\{g(s_k) - f(s_k)\}$.
(*g maximization*) Search for $s_k \in X$ of maximal $g(s_k)$.

Considering the schema representation, the *fg-fitness* evaluation increases as the number of labels # decreases, and therefore structures have higher *fg-fitness* evaluation than schemata. To attain these purposes, a problem to be solved using *CGA* is modeled as the following *Bi-objective Optimization Problem (BOP)*:

$$\text{v(BOP)} = Min\{g(s_k) - f(s_k)\} \quad (1)$$

$$Max\{g(s_k)\} \quad (2)$$

$$\text{subject to} \quad g(s_k) \geq f(s_k),\ s_k \in X \quad (3)$$

Functions f and g must be properly identified to represent the optimization objectives of the problems at issue. The *fg-fitness* process is particularized in the following for the *MCLP*.

In general terms, after the formation of clusters $C_1(s_k), C_2(s_k), ..., C_p(s_k)$ the function f and the function g are computed as follows:

$$f(s_k) = \sum_{i \in I} D_i y_j \; when \; y_j = 1 \; if \; d_{ij} \leq S, j \in \{seed\, vertices\},\ y_j = 0\ ,\ otherwise \; and \; g(s_k) = \sum_{i \in I} D_i\ ;$$

where D_i is the population demand of area i, I={seed and non-seed vertices}, s_k may be a structure or schema and the seeds are the located facilities. The first objective in the *BOP* (interval minimization) assures a maximal covering and the second guides the evolution process to transform schemata in structures.

2.3 The evolution process

The evolution process in *CGA* is conducted to accomplish the objectives (*interval minimization and g maximization*) of the *BOP*. At the beginning of the process, the following two *expected values* are given to these objectives: a non-negative real number $g_{max} > Max\,\{g(s_k)\}\ where\ s_k \in X$, that is an upper bound to $g(s_k)$, for each $s_k \in X$; and the interval length dg_{max}, obtained from g_{max} using a real number $0 < d \leq 1$.

The evolution process is then conducted considering an adaptive rejection threshold, which contemplates both objectives in the *BOP*. Given a parameter $\alpha \geq 0$,the expression

$$g(s_k) - f(s_k) \geq dg_{max} - \alpha.d[g_{max} - g(s_k)] \tag{4}$$

presents a condition for rejection from the current population of a schema s_k. The right hand side of (4) is the threshold, composed of the expected value to the interval minimization dg_{max}, and the measure $[g_{max} - g(s_k)]$, that shows the difference of $g(s_k)$ and g_{max} evaluations. Parameter α is related to time in the evolution process. Considering that the good schemata need to be preserved for recombination, the *evolution parameter* α starts from 0, and then increases slowly, in small time intervals, from generation to generation. The population at the evolution time α, denoted by P_α, is dynamic in size according to the value of the adaptive parameter α, and can be emptied during the process.

2.4 Selection, recombination and mutation

The population is kept ordered according to the "completeness" of the schema, i.e., the number of labels #s, and the schema fg-fitness.

The method used for selection takes the first schema from an initial (best) part of the population (schema base) and the second one from the whole population (schema guide). Before recombination, the first schema is complemented to generate a structure representing a feasible solution, i.e., all #'s are replaced by 0's. This structure undergoes mutation and is compared to the best one found so far. Only the best one is kept along the process. The recombination merges information from both selected schemata, but preserves the number of seeds of the new generated schema or structure. If it is a new schema then it is inserted into the population, otherwise it undergoes mutation and is compared to the best structure found so far.

The recombination is best described in the following. The assignment operations must be performed in that order.

Recombination:

$$\begin{array}{l}
s_{j(base)} = \#\ and\ s_{j(guide)} = \#\ then\ s_{j(new)} = \# \\
s_{j(base)} = 1\ and\ s_{j(guide)} = 1\ then\ s_{j(new)} = 1 \\
s_{j(base)} = 0\ and\ s_{j(guide)} = 0\ then\ s_{j(new)} = 0 \\
s_{j(base)} = 1\ and\ s_{j(guide)} = \#\ then\ s_{j(new)} = 1 \\
s_{j(base)} = 0\ and\ s_{j(guide)} = \#\ then\ s_{j(new)} = 0 \\
s_{j(base)} = \#\ and\ s_{j(guide)} = 0\ then\ s_{j(new)} = 0 \\
s_{j(base)} = \#\ or\ 0\ and\ s_{j(guide)} = 1\ then \\
\qquad s_{j(new)} = 1\ and\ s_{k(new)} = 0\ for\ some\ s_{k(new)} = 1 \\
s_{j(base)} = 1\ and\ s_{j(guide)} = 0\ then \\
\qquad s_{j(new)} = 0\ and\ s_{k(new)} = 1\ for\ some\ s_{k(new)} = 0
\end{array}$$

The mutation process (presented below) used is a local search technique that implements successive changes in the seed position inside each cluster $C_1(s_k), C_2(s_k), ..., C_p(s_k)$ followed by cluster reconstruction made by vertices covering, recalculates $f(s_k)$ and reiterates until no more improvement in $f(s_k)$.

```
While[ $f(s_k)$ increases]
    For $j = 1, ..., p$
        Interchange seed and non-seed nodes in cluster $C_j(s_k)$;
        Calculate the corresponding value $f(s)$ of the best reallocation;
        If $f(s) > f(s_k)$
            Update the seed node for cluster $C_j(s_k)$;
            Set $s_k = s$
        End_If
    End_For
End_While
```

At each generation, after new schemata insertion, the population is scanned to remove all schemata satisfying the condition (4).

3 Computational Tests

For the computational tests we used a real data collected at the central area of the São José dos Campos city (Brazil) using the Geographical Information System (*GIS*) *ArcView*, where three instances 324, 402, 500 are created. Each point is located on a block which presents a demand and is also a possible place to locate seeds. We simulated the installation of radio antennas for Internet service, with short range values (800 m). These instances are available at http://www.lac.inpe.br/∼lorena/instancias.html. Moreover, we used the distance matrices of the 100 and 150 vertex network of Galvão and Revelle [4]. The demand values used were not identical, but generated in the same way: the population of each node were sampled from a uniform distribution in the range [20,30] for the 100-vertex network (service distance equal to 80 m) and from a normal distribution with mean equal to 80 and standard deviation equal to 15 for the 150-vertex network (service distance equal to 95). The algorithm described is coded in C and the tests made on a PC Pentium II 233 MHz. The *CGA* parameters in this case are:

1. For all computational tests, an initial population was randomly created with 20% of symbols 0 and exactly p (number of seeds) symbols 1;

2. The values for parameters $d = 0.1$ and the α *increment* $= 0.01$ are found after an initial set of 10 runs, observing the best results. The upper bound $g_{max} = 2 * g(s_l)$, where s_l is any structure.

The results for the best covers over 20 runs are presented in Table 1. The columns show the problems size and number of seeds (3 and the number assuring complete covering for the real data and 12 and 7 seeds for the other instances), the total population covered, the covering percentage, parameters d and the α increment and the computational times (seconds).

The computational times can be considered expensive but in accordance with *GAs* approaches. The cover found in problem 8 is better than the one of Galvão et. al [3].

Problem	Size	Pop.Attended	Cov(%)	α	d	Time(s)
1	324x3	11604.00	95.49	0.01	0.1	1871
2	324x5	12152.00	100.00	0.01	0.1	5485
3	402x3	14690.00	91.90	0.01	0.1	1851
4	402x6	15984.00	100.00	0.01	0.1	3502
5	500x3	15730.00	79.82	0.01	0.1	982
6	500x8	19707.00	100.00	0.01	0.1	7695
7	100x12	2515.00	100.00	0.01	0.1	362
8	150x7	11085.81	94.09	0.01	0.1	720

Table 1: Results for real data and Galvão and Revelle [4] instances

4 Comments and conclusion

In this work we applied the *CGA* to the *MCLP*. The *MCLP* was considered as a clustering problem in graphs following the Lorena and Furtado approach [6]. The computational results showed that this approach can be useful to the analysis of spatially distributed data. The utilization of a *GIS* database permitted the evaluation with real word data on different scenarios, which can be helpful to decision makers in public and private sectors.

Acknowledgments: The first author acknowledges Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES for financial suport. The second author acknowledges Fundação para o Amparo a Pesquisa no Estado de S. Paulo - FAPESP (proc. 99/06954-7) and Conselho Nacional de Desenvolvimento Cientifico e Tecnológico - CNPQ (proc. 300837/89-5) for partial financial support.

References

[1] C.H. Chung. Recent application of the maximal covering location problem (mclp) model. *J. Oper. Res. Soc.*, 37:735–746, 1986.

[2] R.L. Church and C.S. Revelle. The maximal covering location problem. *Papers of the Regional Science Association*, 32:110–118, 1974.

[3] R.D. Galvão, L.G.A. Espejo, and B. Boffey. A comparison of langrangean and surrogate relaxations for the maximal covering location problem. *European J. Oper. Res.*, 124:377–389, 2000.

[4] R.D. Galvão and C. Revelle. A lagrangean heuristic for the maximal covering location problem. *European J. Oper. Res.*, 88:114–123, 1996.

[5] J.H. Holland. *Adaptation in natural and artificial systems.* MIT Press, 1975.

[6] L.A.N. Lorena and J.C. Furtado. Constructive genetic algorithm for clustering problems. *Evolutionary Computation*, 2000. to appear. Available from http://www.lac.inpe.br/∼lorena/cga_cluster.PDF.

[7] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Spring-Verlag, Berlin, 1996.

[8] Z. Michalewicz and D.B. Fogel. *How to solve it: Modern Heuristics.* Spring-Verlag, Berlin, 2000.

[9] D. Shilling, V. Jayaraman, and R. Barkhi. A review of covering problems in facility location. *Location Science*, 1(1):25–55, 1993.