Efficient Algorithms for Regionalization: an Approach Based on Graph Partition

Gustavo Silva Semaan [*] José André de Moura Brito [†] Luiz Satoru Ochi [*]

[*] Instituto de Computação - Universidade Federal Fluminense, IC-UFF
Rua Passo da Pátria 156 - Bloco E - 3^o andar, São Domingos, CEP: 24210-240, Niterói, RJ, Brasil
{gsemaan, satoru}@ic.uff.br

[†] Escola Nacional de Ciências Estatísticas - Instituto Brasileiro de Geografia e Estatística, ENCE-IBGE
Rua André Cavalcanti 106, sala 403, CEP: 20231-50, Rio de Janeiro, RJ, Brasil
jose.m.brito@ibge.gov.br

ABSTRACT

This paper proposes new approaches based on the GRASP and Evolutionary algorithms for the resolution of a specific regionalization problem. This problem can be mapped on a capacity and connectivity graph partition problem. A review of literature showing that the algorithms work only with the edges of the Minimum Spanning Tree is presented. In this case, the algorithms act on the original graph, in order to increase the possibilities of vertex migration. Results obtained from the application of such algorithms over a set of real data suggested that the use of original graphs through them is a new efficient way to solve this problem.

1. INTRODUCTION

According to [1, 2], regionalization is a clustering procedure applied to spatial objects with a geographic representation, which groups them into homogeneous contiguous regions and Cluster Analysis is a multivariate technique used to group objects together based on a selected similarity measure, in such way that objects in the same cluster are very similar and objects in different clusters are quite distinct [3].

Considering a given set with n objects $X = \{x_1, .., x_n\}$, it must extract partitions from the set X in k different clusters C_i, respecting the following three conditions:

$$\bigcup_{i=1}^{k} C_i = X$$

$$C_i \neq \emptyset, 1 \leq i \leq k$$

$$C_i \cap C_j = \emptyset, 1 \leq i, j \leq k, i \neq j$$

The cluster analysis is a fundamental technique to experimental sciences in which the classification of elements into groups is desirable . As examples of these fields it is possible to cite: biology, medicine, economy, psychology, marketing, statistic among others [4].

2. GRAPH PARTITION PROBLEM

Several clustering problems can be mapped on graph partition problems.This consists in grouping the vertexes of the graphs in different subsets (clusters), according to their similarities, by using a fitness function [1, 5, 6]. Moreover, this regionalization problem considers the following restrictions:

- *Connectivity*: the vertexes grouped in each cluster must be connected.
- *Minimum Capacity*: associated total to one of the variables must be higher than minimum capacity submitted as parameter.

The high combinatorial possibilities of the clustering problems suggests the use of metaheuristic algorithms [7]. This algorithm can reach a typical optimal solution which is very close to global solution, in some cases the global optimal, in a reasonable amount of time. So, papers about clustering problems, including graph partition problem that consider additional restrictions such as connectivity and capacity had been widely reported in literature.

Some Groups [8, 9] had proposed heuristics algorithms for the capacity clustering problem, while others [1, 2, 10] had suggested algorithms for the regionalization problem, in which the connectivity restriction was considered (Automatic Zoning Procedure - AZP and the Spatial 'K'luster Analysis by Tree Edge Removal - SKATER).

The problem presented in this paper considers both connectivity and capacity restrictions into partition graph problem. It is important to underline that, excepting the AZP, the other works referenced that considered the connectivity restriction were based on Minimum Spanning Tree (MST) Partition Method. This method is composed by two steps:

1. Construction of a MST from the graph which represents the problem.
2. Formation of sets of clusters through of partitioning of MST.

According to the connectivity restriction, a natural solution for the problem will consist of building a MST T from G, respecting the smaller values of d_{ij} (1).

$$d_{ij} = \sqrt{\sum_{s=1}^{p} (x_i^s - x_j^s)^2} \qquad (1)$$

In this way, these areas are geographically immediate neighbors, and homogeneity, regarding a set of p variables associated to populational and environmental known characteristics. These variables, which will be represented by x^s, $s = \{1,..,p\}$, are also called indicators (associated variables to each vertex).

Considering these indicators and using the distances d_{ij} between i and j neighbors vertexes are calculated. The distances d_{ij} represent the homogeneity degree, i.e., the proximity among values from p variables associated to all vertexes to be aggregated.

Once provided one tree T and a number k of partitions (cluster to be generated), it is possible to extract $(k-1)$ edges from T, defining, this way, a set of K subtrees T_j, $j=\{1,.., k\}$.Each one of these subtrees will be associated to one cluster.

The connectivity property can be observed in each of the subtrees (clusters). Thus, the solution for the problem will consist of partitioning T in k subtrees T_j, $j=\{1,.., k\}$ associated to cluster what satisfies the capacity restriction and results in the lower possible value for a fitness function(2).

$$f(T) = \sum_{j=1}^{p} \sum_{i=1}^{n} (x_{ij} - \overline{x_j})^2 \tag{2}$$

The case of AZP was based on the spatial object neighbor structure to assure the connectivity restriction and acts, basically, on the migration of the objects in order to minimize a fitness solution.

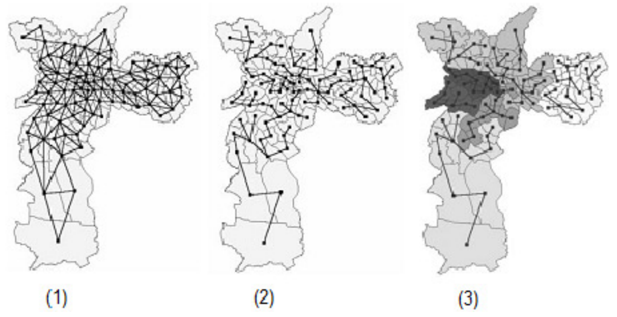

Figure 1: Adjacency relations between objects [1].

According to 1, follows the descriptions of the items: (1) connectivity graph, (2) minimum Spanning Tree and (3) an example of solution.

3. PROPOSED ALGORITHMS

Review of literature showed that the proposed algorithms work only on the edges of MST. In order to increase the possibilities of vertex migration this work presents new heuristic algorithms that act with the original submitted graph of the problem. This proposal enables and facilitates the formation of not only feasible, which the restriction of capacity is respected, but also better quality solutions.

According to [6], a good data structure for the problem is extremely important to the algorithms performance and it can be decisive for a fast convergence and quality of the obtained solutions. The *group-number* structure was used to representation of the solution, where the index of vector represents the vertex of the graph and its content represents the cluster to which the vertex belongs (also used by [5, 6, 11]).

The proposed approach consists in creating solutions using the MST Partition Method through the constructive heuristics, and so, refining its using local search procedures. It was used versions of local search that consider the original graph, and not only the MST built.

3.1. Constructive Heuristics

Two versions of constructive heuristics were proposed, assuring the connectivity restriction through MST Partition Method, both considering the concepts of GRASP Metaheuristic (*Greedy Randomized Adaptive Search Procedures* [12]).

While a first version worked aiming to build feasible solutions, which the restriction of capacity is respected, the second version acted in order to minimize the fitness solution, independently of the restriction of capacity.

Both versions act to generate k partitions, removing (k − 1) edges from T, since the hierarchical division strategy was used and, initially, all the vertexes belong to the same cluster.

The Constructive Heuristic 1 (CH1) was proposed by [11] and consists in, after the selection of the cluster (associated with a subtree T_i) that must be partitioned (what have the high fitness function), to evaluate all the possibilities of edge removal in order to minimize the fitness function. This way, must be removed the edge of high value of (3) of the subtree T_i, generation two new subtrees T_i^1 and T_i^2.

$$C_{edge} = f(T_i) - (f(T_i^1) + f(T_i^2)) \tag{3}$$

Although it is a greedy procedure which has an expensive computational cost, it was applied on the building of the initial solution for the proposed algorithm. In order to make this algorithm semi-greedy, it was used a Restricted Candidate List (RCL), which the α high edges (according C_{edge} value) are selected and, one of them is randomly selected, aiming to divide the selected cluster.

The Constructive Heuristic 2 (CH2) was based on the CH1 but, in this version, intending to obtain valid solutions. In this case, the selection of the cluster that must be partitioned occurs by capacity criteria, in which the cluster with higher capacity must be selected. Moreover, the algorithm is also semi-greedy and a RCL was used. In order to build valid solutions, the CH2 acts dividing the selected cluster C_w (subtree T_w) in the clusters C_{w1} and C_{w2} and, afterwards, one of them must have its capacity minimized and the capacity criteria respected.

3.2. Local Search Procedures

Six versions of Local Search (LS) were used considering:

- *MST*: only the edges of the MST built.
- *Original Graph*: all edges from the original submitted graph.
- *Feasible Solutions*: construction of valid solutions.
- *Better Solutions*: to minimize the fitness solution, independent of the restriction of capacity.

Table 1 ilustrates the distributions of the Local Search versions among the considering properties.

Property	LS1	LS2	LS3	LS4	LS5	LS6
MST	x			x	x	
Original Graph		x	x			x
Feasible Solutions	x	x			x	
Better Solutions			x	x		x

Table 1: Properties by Local Search versions.

Descriptions of the Local Search versions:

- *LS1*: uses the edges that were selected during the cluster partition. Basically, the procedure verifies if one and only one cluster associated to vertexes of the edge is penalized (if it has capacity less than the minimum capacity). In this case, the vertex is migrated to this cluster, aiming to regenerate the solution.
- *LS2*: realizes migrations of vertexes based on the original submitted graph of the problem, aiming to regenerate the infeasible solutions.
- *LS3*: realizes migrations of vertex based on the original submitted graph of the problem aiming to minimize the fitness' solution.
- *LS4 and LS5*: work joining adjacent clusters in which exists an edge connecting vertexes of this clusters, and after, dividing this cluster using, respectively, the CH1 and CH2 procedures.
- *LS6*: was based on the known clustering algorithm of the literature, the K-Means [13, 4] but, in this case, the restrictions of this problem were considered.

3.3. Additional Comments about the Implementation

This paper proposes Evolutionary Algorithms (EA)[12] that bring together the construtives and local search procedures. It follows the other implemented techniques:

- *Crossover*: the vertexes migration occur by the 1-point type crossover operator. It was necessary to verify if the new solutions have k clusters and if the clusters are connected.
- *Mutation*: it was used random vertex migration, aiming to perturb the solution.
- *Elitism*: The best found solutions are saved and inserted to the next population in order to improve quality by using the others procedures.
- *Minimum Capacity*: Total associated to one of the variables. This value can be either submitted as parameter or calculated at the begin of the algorithm, which β is the fit factor, k a number of clusters, n a number of vertexes and x_i^s the variable s associate with the vertex i (4).

$$Cap_{Min} = (\beta/k).\sum_{i=1}^{n} x_i^s \tag{4}$$

In the experiments, were considered only two versions of EA:

- *EAOG*: Evolutionary Algorithm that consider the original submitted graph. It was used: LS2, LS3, LS6, Elitism, CH1 or CH2.
- *EAMST*: Evolutionary Algorithm that consider only the edges of the MST. It was used: LS1, LS4, LS5, Crossover, Mutation, Elitism, CH1 or CH2.

4. COMPUTATIONAL RESULTS

A real set of twenty six instances from Brazilian Demographic Census (data for public use) was used for the experiments. Moreover, the algorithms presented were coded in Ansi C, running on a Intel Centrino II 2,4 GHz processor and 4GB RAM.

Table 2 presents properties of the used instances, where each vertex is a weigthed area. A weighted area is a small geographical area formed by a mutually exclusive enumeration areas (cluster of census segments), which comprise, each one of them, a set of records of households and people. And the associated variables are: total of houses, total of domiciles, total of person, sum of salaries, sum of time of instruction or study, sum of salary per-capita, average time of instruction or study of the responsible.

Id	|Vertex|	|Edge|	Id	|Vertex|	|Edge|
1	21	58	14	178	791
2	61	286	15	121	567
3	409	2020	16	75	359
4	73	350	17	114	502
5	14	46	18	133	620
6	18	59	19	195	868
7	89	363	20	68	307
8	16	60	21	181	843
9	57	236	22	151	560
10	375	1769	23	86	388
11	179	882	24	155	722
12	74	357	25	461	2385
13	231	1172	26	285	1451

Table 2: Real instances of Brazilian Demographic Census.

Aiming to calibrate the parameters, several preliminary experiments were run based on the selected set of instances. The obtained parameters were: k=3 (clusters), PopulationSize=10 solutions, StopCriteria=100 generations, Crossover=80%, Mutation=5% and α=5. The crossover and mutation have a high probability since its execution is evaluate in order to form only feasible solutions.

Although real applications can define the Minimum Capacity for each instance, in this experiment was fixed $\beta = 30\%$.

In the experiment, each algorithm was executed over the same instance twenty times. The elapsed time and the gap associated with the best known result of the each instance were obtained.

The tables 3 and 4 present, respectively, the best of this results by EA version for each instance and some statistics about this experiment. The EAOG obtained best results for all the instances, however, its average of elapsed time was higher then EAMST versions.

$$Gap(AEGO, EAMST) = 100 * \frac{|f_{AEGO} - f_{AEMST}|}{f_{AEGO}} \tag{5}$$

Id	Gap	Id	Gap	Id	Gap
1	26.97	10	43.33	19	54.08
2	7.1	11	61.85	20	16.44
3	5.82	12	40.31	21	41.86
4	20.3	13	51.23	22	39.05
5	11.71	14	91.09	23	60.6
6	3.97	15	65.76	24	48.96
7	78.84	16	35.38	25	26.46
8	17.44	17	56.49	26	56.25
9	59.59	18	84.02		

Table 3: Gap between EAOG and EAMST.

In order to analyze the results, three categories were created according to the Gap values of the best solution known: Best (Gap = 0%), Interesting (Gap $\leq$ 5%) and Bad ($Gap > 70\%$).

Average Time	EAOG	269 seconds
	EAMST	133 seconds
Gap (EAOG, EAMST)	Min	3.97%
	Max	91.09%
	Mean	42.49%
	Median	42.59%
Gap [Best Known reference]	EAOG	4.00%
	EAMST	51.00%

Table 4: Statistics.

The table 5 presents the results by categories.

Categories	EAOG	EAMST
Best	40%	12%
Interesting	60%	17%
Bad	0%	29%

Table 5: Results by categories.

Since the AEOG reached best results but its elapsed time was higher than of AEMST, both algorithms were submitted to a new experiment. They were run one hundred times, over three among the bigger selected instances and, in this experiment, the StopCriteria was a maximum time (300 seconds) or the solution reach the target value, submitted as parameters.

In this experiment all the AEOG executions reached the target, while the AEMST had probabilities of 52%, 55% and 38% for the instances 4, 13 and 22, respectively.

Despite the algorithm had obeyed the stipulated processing time, the EAMST continued limited in best local solutions, while the EAOG obtained new different solutions, that could not be formed through the only MST method. Moreover, the AEOG reached the target of the instances 4, 13 and 22 at 40, 10 and 10 seconds, respectively.

5. CONCLUSIONS

In this paper two versions of constructive heuristics were proposed, both considering the concepts of GRASP Metaheuristic. Afterwards, six local search procedures were used aiming to refine the solutions, in order to increase de solutions' quality or regenerate infeasible solutions.

Two Evolutionary Algorithms were presented, bring together the construtives and local search procedures: The EAOG (based on the Original Graphs) and EAMST (based only on edges of MST).

It was possible to confirm that the procedures that acted with the original submitted graph increase the possibilities of vertex migration and thus facilitated the formation of both valid as better quality solutions.

The computational results showed that the use of Constructive Heuristics that consider only edges of MST together a local search procedures and the use of Original Graphs are an interesting alternative to solve this problem, improving both the solution's quality as the quantity of formation of valid solutions.

These results indicate that the proposed heuristics are an efficient way to solve this problem. Besides, as another ways to solve it we can cite: the use of Pathrelinking in order to integrate intensification and diversification in search for new best solutions [12]; to develope and analyze the use of other metaheuristics, such as: Iterated Local Search (ILS), Variable Neighborhood Search (VNS), Tabu Search or a hybrid heuristic version [12].

6. ACKNOWLEDGMENTS

To all the teachers and students of the Computer Institute at UFF (http://www.ic.uff.br) and CAPES (http://www.capes.gov.br) for the financial support.

7. REFERENCES

[1] R. M. Assunção, M. C. Neves , G. Câmara, C. Freitas, "Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees," *International Journal of Geographical Information Science*, vol. 20, no. 7, 2006.

[2] M.J. Smith, M. F. Goodchild, P. A. Longley, *Geospatial Analysis : a Comprehensive Guide to Principles, Techniques and Software Tools*. Troubadour Publishing Limited, 2009.

[3] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2006.

[4] H. C. Romesburg, *Cluster Analysis for Researchers*. Lulu Press, 2004.

[5] C. R. Dias, L. S. Ochi, "Efficient evolutionary algorithms for the clustering problems in directed graphs," in *Proc. of the IEEE Congress on Evolutionary Computation (IEEE-CEC)*, Canberra, Austrlia, 2003, pp. 983–988.

[6] D. Doval, S. Mancoridis, B. S. Mitchell, "Automatic clustering of software systems using a genetic algorithm," in *Proc. of the Int. Conf. on Software Tools and Engineering Practice*, Pittsburgh, USA, 1999, pp. 73–81.

[7] P. Hansen, B. Jaumard, "Cluster analysis and mathematical programming," *Mathematical Programming*, vol. 79, pp. 191–215, 1997.

[8] S. W. Scheuerer, "A scatter search heuristic for the capacitated clustering problem," *European Journal of Operational Research*, vol. 169, 2006.

[9] H. M. Shieh, M. D. May, "Solving the capacitated clustering problem with genetic algorithms," *Journal of the Chinese Institute of Industrial Engineers*, vol. 18, 2001.

[10] R. M. Assuncao, J. P. Lage, A. E. Reis, "Analise de conglomerados espaciais via arvore geradora minima," *Revista Brasileira de Estatstica*, 2002.

[11] G. S. Semaan, L. S. Ochi, J. A. M. Brito, "An efficient evolutionary algorithm for the aggregated weighting areas problem," in *International Conference on Engineering Optimization*, 2008.

[12] F. Glover, *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003.

[13] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.