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° andar, São Domingos, CEP: 24210-240, Niterói, RJ, Brasil

† 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

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 \( X = \{x_1, \ldots, x_n\} \), it must extract partitions from the set \( X \) in \( k \) different clusters \( C_i \), respecting the following three conditions:

\[
\bigcup_{i=1}^{k} C_i = X \quad C_i \neq \emptyset, 1 \leq i \leq k \quad C_i \cap C_j = \emptyset, 1 \leq i, j \leq k, i \neq j
\]

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 [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 \( d_{ij} \) [11].

\[
d_{ij} = \sqrt{\sum_{t=1}^{p} (x_t^i - x_t^j)^2}
\]

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 \( x^s \), \( s = \{1, \ldots, p\} \), are also called indicators (associated variables to each vertex).
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 [5,5,11]).

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 Tj), 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 Tj, generation two new subtrees Tj1 and Tj2.

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.

\[
f(T) = \sum_{i=1}^{n} \sum_{j=1}^{p} (x_{ij} - x_{ij})^2
\]  

(2)

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.
- **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 \(\beta\) 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 \(i\) \[4\].

\[
\text{CapMin} = (\beta/k) \cdot \sum_{i=1}^{n} x^i_s
\]

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

- **EAOG**: Evolutionary Algorithm that consider only the edges of the MST. 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 weighed 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.

<table>
<thead>
<tr>
<th>Id</th>
<th>Vertex</th>
<th>Edge</th>
<th>Id</th>
<th>Vertex</th>
<th>Edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21</td>
<td>58</td>
<td>14</td>
<td>178</td>
<td>791</td>
</tr>
<tr>
<td>2</td>
<td>61</td>
<td>286</td>
<td>15</td>
<td>121</td>
<td>567</td>
</tr>
<tr>
<td>3</td>
<td>409</td>
<td>2020</td>
<td>16</td>
<td>75</td>
<td>339</td>
</tr>
<tr>
<td>4</td>
<td>73</td>
<td>350</td>
<td>17</td>
<td>114</td>
<td>502</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>46</td>
<td>18</td>
<td>133</td>
<td>620</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>59</td>
<td>19</td>
<td>195</td>
<td>868</td>
</tr>
<tr>
<td>7</td>
<td>89</td>
<td>363</td>
<td>20</td>
<td>68</td>
<td>307</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>60</td>
<td>21</td>
<td>181</td>
<td>843</td>
</tr>
<tr>
<td>9</td>
<td>57</td>
<td>236</td>
<td>22</td>
<td>151</td>
<td>560</td>
</tr>
<tr>
<td>10</td>
<td>375</td>
<td>1769</td>
<td>23</td>
<td>86</td>
<td>388</td>
</tr>
<tr>
<td>11</td>
<td>179</td>
<td>882</td>
<td>24</td>
<td>155</td>
<td>722</td>
</tr>
<tr>
<td>12</td>
<td>74</td>
<td>357</td>
<td>25</td>
<td>461</td>
<td>2385</td>
</tr>
<tr>
<td>13</td>
<td>231</td>
<td>1172</td>
<td>26</td>
<td>285</td>
<td>1451</td>
</tr>
</tbody>
</table>

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 \(\alpha=5\). 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.

\[
\text{Gap}(\text{EAMST}) = 100 \cdot \frac{|f_{\text{AEGO}} - f_{\text{AEGO}}|}{f_{\text{EAMST}}} \quad (5)
\]

Table \(3\): Gap between EAOG and EAMST.

<table>
<thead>
<tr>
<th>Id</th>
<th>Gap</th>
<th>Id</th>
<th>Gap</th>
<th>Id</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26.97</td>
<td>10</td>
<td>43.33</td>
<td>19</td>
<td>54.08</td>
</tr>
<tr>
<td>2</td>
<td>7.11</td>
<td>11</td>
<td>61.85</td>
<td>20</td>
<td>16.44</td>
</tr>
<tr>
<td>3</td>
<td>5.82</td>
<td>12</td>
<td>40.31</td>
<td>21</td>
<td>41.86</td>
</tr>
<tr>
<td>4</td>
<td>20.3</td>
<td>13</td>
<td>51.23</td>
<td>22</td>
<td>39.05</td>
</tr>
<tr>
<td>5</td>
<td>11.71</td>
<td>14</td>
<td>91.09</td>
<td>23</td>
<td>60.6</td>
</tr>
<tr>
<td>6</td>
<td>3.97</td>
<td>15</td>
<td>65.76</td>
<td>24</td>
<td>48.96</td>
</tr>
<tr>
<td>7</td>
<td>78.84</td>
<td>16</td>
<td>35.38</td>
<td>25</td>
<td>26.46</td>
</tr>
<tr>
<td>8</td>
<td>17.44</td>
<td>17</td>
<td>56.49</td>
<td>26</td>
<td>56.25</td>
</tr>
<tr>
<td>9</td>
<td>59.39</td>
<td>18</td>
<td>84.02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 ≤ 5%) and Bad (Gap > 70%).
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).

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 EAOG reached the target of the instances 4, 13 and 22 at 40, 10 and 10 seconds, respectively.

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 Stop Criteria 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 EAO (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
develop 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


