Lagrangean Relaxation with Clusters for Point-Feature Cartographic Label Placement Problems

Glaydston Mattos Ribeiro and Luiz Antonio Nogueira Lorena[1]

LAC – Computer and Applied Mathematics Laboratory

INPE – Brazilian Space Research Institute

12.227-010, São José dos Campos – SP

Tel: +55 12 3945-6555 FAX: +55 12 3945-6357

E-mail: {glaydston, lorena}@lac.inpe.br

Abstract

This paper presents two new mathematical formulations for the Point-Feature Cartographic Label Placement Problem (PFCLP) and a new Lagrangean relaxation with clusters (LagClus) to provide bounds to these formulations. The PFCLP can be represented by a conflict graph and the relaxation divides the graph in small sub problems (clusters) that are easily solved. The edges connecting clusters are relaxed in a Lagrangean way and a subgradient algorithm improves the bounds. The LagClus was successfully applied to a set of instances up to 1000 points providing the best results of those reported in the literature.

Keywords: Lagrangean Relaxation, Integer Programming, Label Placement.

[1] Corresponding author

1. Introduction

The point-feature cartographic label placement problem (PFCLP) can be considered as a combinatorial optimization problem. The problem is to place point labels in positions in a way that a map without overlaps is obtained (See Figure 1).

Cartographic standardization [4] determines possible locations for the labels. Defining these positions, this problem can be modeled as a combinatorial optimization problem. Figure 2 shows a set of 8 possible positions for a label, which are called candidate positions. The numbers indicate the cartographic preference, and the upper right is the best cartographic position.

Consider the problem with 4 candidate positions for each point shown in Figure 3(a). It can be easily represented by a conflict graph. Let N be the number of points that must be labeled and P the number of candidate positions for each point. $G=\{V,A\}$ is the corresponding conflict graph, where $V=\{v_1, v_2, \ldots,v_{N*P}\}$ is the set of candidate positions (vertices) and $A=\{(v_i, v_j): i, j\ \hat{I}\ V, i^{1}j\}$ the conflicts (edges). Figure 3(b) presents the conflict graph obtained from Figure 3(a), and Figure 3(c) shows the optimal solution for this problem. Usually in the literature, proportions of conflict free labels assess the quality of labeling. In the case shown at Figure 3(c), we have 100% of conflict free labels.

The approaches studied in literature have different but connected objectives. The PFCLP can be modeled as a Maximal Independent Vertex Set Problem (MIVSP) [25] or as a Maximum Number of Conflict Free Labels Problem (MNCFLP) [4]. Both approaches count the final number of positioned conflicts free labels, but in MIVSP points with inevitable overlaps are not labeled, while all points must be labeled in MNCFLP.

The MNCFLP is more useful under the cartographic point of view than the MIVSP, but the map visibility is not fully explored. Figure 4 shows two possible solutions for a problem with four points. While both solutions are equivalent for MNCFLP presenting all labeled points in conflict, solution (b) has better visibility than solution (a). Besides, if we only count the number of conflicts (edges) in their graphs, solution (b) is better than (a).

Considering the visibility questions in a map, the aim of this paper is to propose a new approach for the PFCLP, contributing with two integer linear programming models for the Minimum Number of Conflicts Problem (MNCP), and also presenting a Lagrangean heuristic that is applied after decomposition of the conflict graph in clusters, obtaining better results than those reported in the literature for a set of instances up to 1000 points. The MNCP, like the MNCFLP, labels all points in a map.

The rest of the paper is organized as follows: in the next section, a brief review is shown about PFCLP, followed by the two proposed mathematical models. In Section 4 the relaxations are shown, followed by the computation results and conclusions.

2. Literature Review

Considering the PFCLP as a Maximal Independent Vertex Set Problem (MIVSP), a substantial research exists in algorithms and techniques to reduce the number of the generated constraints. The MIVSP has several applications in different fields such as in location of military defenses [3], Cut and Packing [1], pallet loading [7], DNA sequence [15], allocation models [9], anti-covering [19], forest planning [20] [6], harvest scheduling [10] and cellular networks [2].

Specifically considering the MIVSP as a PFCLP, Zoraster [30] [31] [32] formulated mathematically the PFCLP working with conflict constraints and dummy candidate positions of high cost if candidate positions could not be labeled. He also proposed a Lagrangean relaxation for the problem and obtained some computational results on small-scale instances.

Strijk et al [25] proposed other mathematical formulation exploring some cut constraints. These cuts are based on cliques and appeared before in the works of Moon and Chaudhry [18] and Murray and Church [21]. They also applied and proposed many heuristics: Simulated Annealing, Diversified Neighborhood Search, *k-opt* and Tabu Search. The last one showed the better results for their instances.

The Maximum Number of Conflict Free Labels Problem (MNCFLP) was examined in several papers. Hirsh [13] developed a Dynamic Algorithm of label repulsion, where labels in conflicts are moved trying to remove a conflict. Christensen et al [4] [5] proposed an Exhaustive Search Approach, alternating positions of the labels that were previously positioned. Christensen et al [5] also proposed a Greedy Algorithm and a Discrete Gradient Descent Algorithm. These algorithms have difficulty of escaping from local maximum. Verner et al [26] applied a Genetic Algorithm with mask such that if a label is in conflict the changing of positions are allowed by crossover operators.

Yamamoto et al [27] proposed a Tabu Search Algorithm that provides very good results when compared with the literature. Schreyer and Raidl [24] applied Ant Colony System but the results found were not interesting when compared to the ones obtained by Yamamoto et al [27]. Yamamoto and Lorena [28] developed an exact algorithm for small instances of MNCFLP and applied the Constructive Genetic Algorithm (CGA) proposed by Lorena and Furtado [17] to a set of large-scale instances. The exact algorithm was applied to instances of 25 points and the CGA was applied to instances up to 1000 points, providing the best results of the literature.

3. Mathematical Formulation Based on Candidate Positions

The first mathematical formulation proposed for the Minimum Number of Conflicts Problem (MNCP) looks at the candidate positions to construct the conflict graph The objective is to minimize the number of conflicts considering that for each point i correspond a number P_i of candidate positions. Each candidate position is represented by a binary variable $x_{i,j}$, $i \in \{1,\ldots,N\}$, $j \in \{1,\ldots,P_i\}$, and N is the number of points that will be labeled. If $x_{i,j} = 1$ the candidate position j of the point i will be used (it will receive the label of point i), otherwise, $x_{i,j} = 0$. Besides, for each possible candidate position of point i is associated a cost (a penalty) represented by $w_{i,j}$.

For each candidate position $x_{i,j}$ corresponds a set $S_{i,j}$ of index pairs of candidate positions conflicting with $x_{i,j}$. $S_{i,j}$ is the set of index pairs (k,t) of candidate positions $x_{k,t}$ conflicting with $x_{i,j}$. For all $(k,t) \in S_{i,j}$, where $k \in \{1,\ldots,N\}:k>i$ and $t \in \{1,\ldots,P_k\}$, corresponds a binary variable $y_{i,j,k,t}$ representing the conflict (an edge in the conflict graph G).

Now, considering the information above, the objective function of the Minimum Number of Conflicts Problem (MNCP1) can be represented by:

$$v(MNCP1) = Min \sum_{i=1}^{N} \sum_{j=1}^{P_i} \left(w_{i,j} x_{i,j} + \sum_{(k,t) \in S_{i,j}} y_{i,j,k,t} \right) \qquad (1)$$

However, for each point i only one candidate position should be selected. Consequently, only one of the candidate positions from P_i will be receiving the value 1. This set of constraints can be written as:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \quad \forall\; i = 1....N \tag{2}$$

Considering the conflicts, placing a label in a candidate position should be taken into account the potential overlaps, and thereby a new constraint set is necessary. This set of constraints considers each position$x_{i,j}$, its respective conflict positions $x_{k,t}$ and one conflict variable $y_{i,j,k,t}$, expressed as:

$$x_{i,j} + x_{k,t} - y_{i,j,k,t} \le 1 \quad \begin{array}{l} \forall\; i = 1...N \\ \forall\; j = 1...P_i \\ (k,t) \in S_{i,j} \end{array} \tag{3}$$

Thus, the first formulation to MNCP is the following binary integer linear programming problem:

$$v(MNCP1) = Min \sum_{i=1}^{N} \sum_{j=1}^{P_i} \left(w_{i,j} x_{i,j} + \sum_{(k,t) \in S_{i,j}} y_{i,j,k,t} \right) \tag{4}$$

Subject to:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \quad \forall\; i = 1...N \tag{5}$$

$$x_{i,j} + x_{k,t} - y_{i,j,k,t} \text{ £ } 1 \qquad \text{"}\ i = 1 \ldots N \quad \text{"}\ j = 1 \ldots P_i \quad (k,t)\ \hat{\text{I}}\ S_{i,j} \tag{6}$$

$$x_{i,j}, x_{k,t}\ and\ y_{i,j,k,t}\ \hat{\text{I}}\ \{0,1\} \qquad \text{"}\ i = 1 \ldots N \quad \text{"}\ j = 1 \ldots P_i \quad (k,t)\ \hat{\text{I}}\ S_{i,j} \tag{7}$$

Constraint (7) ensures that all decision variables of the problem are binaries. When the objective function is minimized the conflict variables should be eliminated or minimized (if elimination is not possible). The formulation (4)-(7) is similar to the one proposed by Zoraster [31], however it allows positioning all labels minimizing the number of conflicts.

This formulation was initially tested using CPLEX 7.5 [14] on a set of standard problems available at http://www.lac.inpe.br/~lorena/instancias.html that are standard sets of randomly generated points: grid size of 792 by 612 units, fixed size label of 30 by 7 units and page size of 11 by 8.5 inch [28]. CPLEX uses fast algorithms and techniques, including cuts, heuristics and a variety of branching and node selection strategies. So, the optimal solution could be found in few seconds for the instances up to 500 points. For the larger instances with 750 and 1000 points, that are approximately solved by Yamamoto et al [27] and Yamamoto and Lorena [28], the optimal solutions were found in 9 of 25 instances in the problems with 750 points, and none for problems with 1000 points. CPLEX was running in several hours until reaching an out of memory state for a 512 MB RAM memory Pentium IV 2.66 GHz machine.

4. Mathematical Formulation Based on Candidate Positions and Points

The second mathematical formulation proposed for the PFCLP considers the conflict graph formed by all candidate positions and its conflicts with points. It was inspired in the work of Murray and Church [21]. Given that for each point *i* only one candidate position will be used (constraints (5)), the conflict constraints (constraints (6)) will represent conflicts of candidate positions and points instead of others candidate positions.

In addition to the variables and sets above mentioned, let $C_{i,j}$ be a set with all points that contain candidate positions in conflict with the candidate position $x_{i,j}$, and $y_{i,j,c}$ a conflict variable between the candidate position $x_{i,j}$ and the point $c \in C_{i,j}$: $c > i$. So, the constraints (6) can be reformulated by the following constraints:

$$\left|C_{i,j}\right| x_{i,j} + \sum_{(k,t)\in S_{i,j}} x_{k,t} - \sum_{c\in C_{i,j}} y_{i,j,c} \le \left|C_{i,j}\right| \quad \forall\, i = 1 \ldots N \quad \forall\, j = 1 \ldots P_i \qquad (8)$$

As the constraints defined in (8) considers conflict variables that indicate conflicts between candidate positions and points, the objective function (4) must be replaced by:

$$v(MNCP2) = Min \sum_{i=1}^{N} \sum_{j=1}^{P_i} \left(w_{i,j} x_{i,j} + \sum_{c\in C_{i,j}} y_{i,j,c} \right) \qquad (9)$$

Thus, the MNCP can be reformulated as:

$$v(MNCP2) = Min \sum_{i=1}^{N} \sum_{j=1}^{P_i} \left(w_{i,j} x_{i,j} + \sum_{c\in C_{i,j}} y_{i,j,c} \right) \qquad (10)$$

Subject to:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \quad \forall\, i = 1 \ldots N \tag{11}$$

$$\begin{aligned} |C_{i,j}|\, x_{i,j} + \sum_{(k,t)\in S_{i,j}} x_{k,t} - \sum_{c \in C_{i,j}} y_{i,j,c} \le |C_{i,j}| \quad & \forall\, i = 1 \ldots N \\ & \forall\, j = 1 \ldots P_i \end{aligned} \tag{12}$$

$$\begin{aligned} x_{i,j}, x_{k,t} \; and \; y_{i,j,c} \in \{0,1\} \quad & \forall\, i = 1 \ldots N \\ & \forall\, j = 1 \ldots P_i \\ & c \in C_{i,j} \end{aligned} \tag{13}$$

Table 1 reports the average number of constraints generated by MNCP1 and MNCP2 formulations for the instances proposed by Yamamoto and Lorena [28], considering 4 candidate positions for each point. In the first column we can see the number of points, followed by the number of instances and the average number of constraints generated by MNCP1 and MNCP2. MNCP2 reduces significantly the number of constraints.

The MNCP2 formulation is also tested running CPLEX on the set of instances proposed in [28], obtaining 12 optimal solutions of 25 for instances with 750 points. It shows that the MNCP2 model appears to be better than MNCP1. But, again with 1000 points, no optimal solution was found in several hours until CPLEX reaches an out of memory state.

So, to find good lower and upper bounds, we applied a Lagrangean heuristic, observing the particular case of $w_{i,j}=1$ in both models. In this case, MNCP1 and MNCP2 have a trivial lower bound equal to *N*, when all points are labeled without conflicts.

4. Lagrangean Relaxation with Clusters

We examine in this section a Lagrangean relaxation formed after the decomposition of the graph *G* in clusters. The relaxation was proposed observing that the conflict graph for PFCLP is usually sparse and well adapted for a previous clustering phase. For example, for MNCP1 model, the graph shown in Figure 5 (a) can be partitioned in two clusters (b). In this partition some constraints represented by edges inter clusters are ignored (c) and the two smaller problems (d) can be independently solved. Zoraster [33] also partitioned the data with other objective in a Simulated Annealing algorithm for solving point feature label placement problems on petroleum industry.

Thus, based on this idea and considering the MNCP1 formulation, we propose a new Lagrangean relaxation with clusters following the steps:

i. Apply a graph partitioning heuristic to divide *G* in *m* parts, forming *m* clusters. The problem now can be written through the objective function defined in (4) subject to (5) and (6), where the conflict constraints (6) can be divided in two groups: one with conflict constraints corresponding to edges intra clusters and other formed by conflict constraints that correspond to edges connecting the clusters.

ii. Two distinct multipliers relax, in the Lagrangean way, the constraints (5) and the conflict constraints corresponding to edges inter clusters.

iii. The resultant Lagrangean relaxation is decomposed in *m* sub-problems and solved. This Lagrangean relaxation will be denoted by LagClus hereafter.

Relaxing constraints (5) the solution cannot be feasible to *P* and the following heuristics CH and IH are used to obtain and improve a feasible solution.

Constructive Heuristic - CH

Fill a feasible solution array with zeroes;

```
For i=1 to N
    Find in relaxed solution all positions different from zero for the point i.
    Select for feasible solution in the position i the candidate position j with smallest number of
    conflicts with elements in feasible solution. In case of tie, select the position corresponding to set
    $S_{i,j}$ with smallest cardinality.
    If none candidate position j for the point i is in relaxed solution, choose the candidate position
    corresponding to the candidate position set $S_{i,j}$ with smallest cardinality.
End For.
```

Improvement Heuristic - IH

```
For each element of feasible solution, store in a conflict array the number of conflicts for each
position.
For i=1 to the length of the conflict array;
    If Conflict array[i] $\neq$ 0
            Seek among the possible candidate positions j, the one that presents the smallest
            number of conflicts with the current feasible solution.
            If there is some candidate position j with the number of the conflicts smaller than Conflict
            array[i], change Feasible Solution [i] with candidate position j.
End For.
```

The Lagrangean sub-problems can be solved by CPLEX in reasonable times. The partitioning of graph *G* was realized using METIS [16], a well-known heuristic for Graph Partitioning Problems. Given a conflict graph *G* and a pre-defined number *m* of clusters, the METIS divides the graph in *m* clusters minimizing the number of edges with terminations in different clusters. Recently Hicks et al [12] found good results applying this technique to Maximum Weight Independent Set Problems.

A subgradient algorithm is used to solve the Lagrangean dual [23]. The subgradient method is similar to the one proposed by Held and Karp [11] and updates the multipliers considering step sizes based on the relaxed solutions and the feasible solutions obtained with the heuristics CH and IH. We implemented the algorithm described by Narciso and Lorena [22].

Now, as the MNCP2 model considers conflicts between candidate positions and points, we apply the LagClus in an alternative mode. Figure 6 shows an example where the graph (b) is obtained from problem (a). We transform it in a point graph and a graph partitioning heuristic is applied (Figure 6 (c)). Starting from (c), we rebuild the original problem (d). At the end, the edges with terminations in different clusters (e) are relaxed in the Lagrangean mode generating smaller sub-problems that can be independently solved (f).

Therefore for the MNCP2, the LagClus follows the steps:

i. Apply a graph partitioning heuristic to divide *G* in *m* parts (G is a graph of conflicts between candidate positions and points, like Figure 6(c)), forming *m* clusters. The problem now can be written through the objective function defined in (10) subject to (11) and (12), where the conflict constraints (12) can be divided in two groups: one with conflict constraints corresponding to edges intra clusters and other formed by conflict constraints that correspond to edges connecting the clusters.

ii. Distinct non-negative multipliers relax in the Lagrangean way the conflict constraints corresponding to edges inter clusters.

iii. The resultant Lagrangean relaxation is decomposed in *m* sub-problems and solved.

Observe that the constraints (5) are not relaxed, so all relaxed solution are feasible to PMNC, and thus we use only the IH heuristic in the subgradient algorithm, that is the same explained before.

5. Computational Results

The computational tests are performed on instances proposed by Yamamoto and Lorena [28] that are available at http://www.lac.inpe.br/~lorena/instancias.html that were used in previous works (See Yamamoto and Lorena [28]). The code in C++ and the tests were done in a computer with Pentium IV 2.66 GHz processor and 512 MB of RAM memory. As done by Zoraster [31], Yamamoto et al [27] and Yamamoto and Lorena [28], for all problems the cartographic preferences were not considered. It allowed us to compare our results to the ones present in the literature considering the cost or penalty equal to 1 for all the candidate positions, being the number of those positions equal to 4: $w_{i,j}=1$ "$i=1\ldots N$ and "$j=1\ldots 4$. We believe that the LagClus can provide better results for 8 candidate positions, but with the corresponding increase in computational times.

Tables 2 and 3 report the average results obtained for MNCP1 and MNCP2 models with CPLEX. The columns are the same in both tables. The first column shows the number of points, followed by the lower bound, upper bound and the gap = *(Lower bound-Upper bound)/Upper bound*100*. The fifth column presents the time in seconds, followed by the number of labels in conflict and by the proportion of free labels. We can see that for problems up to 500 points the results are the same, except the time that was smaller for MNCP2 model. For the others problems (750 and 1000 points), CPLEX obtained better solutions with the MNCP2 than the MNCP1 formulation. It optimally solved 9 of 25 instances with 750 points with MNCP1 and 12 with MNCP2. For problems with 1000 points no optimal solution was found.

Tables 4 and 5 report the average results obtained with LagClus. They have one more column than Tables 2 and 3, which notify the number of clusters used. These numbers were empirically obtained. We can see that the results found for MNCP2 are better than the results for MNCP1. For the instances with 750 points, the LagClus for MNCP2 found 21 optimal solutions of 25 instances while for MNCP1 no optimal solution was found. For problems with 1000 points, the LagClus for MNCP1 and MNCP2 did not found optimal solutions, but the bounds found for MNCP2 are better than those found by the CPLEX.

Trying to obtain optimal solutions for instances with 1000 points, the number of clusters was reduced to 20 for MNCP2, and to see what happens if the number of clusters increases, we make another experiment with 30 clusters. The new results are reported in Table 6. As expected, when the number of clusters is reduced the time increases, but the solutions are better than the ones reported before. The opposite behavior is obtained when the number of clusters is increased. It indicates that with more constrained sub-problems the LagClus has better quality results.

The best results found in this paper are compared to the best results of literature described in the works of Yamamoto et al [27], Yamamoto [29] and Yamamoto and Lorena [28]. Table 7 compares the best average percentages of conflict free labels found using the models proposed (relaxations) with the best results found in the literature. Observe that those approaches have different objectives: the MNCFLP maximizes the number of conflict free labels and MNCP minimize the number of conflicts considering visibility questions, and the LagClus found better results to PFCLP than all reported in the literature considering the number of conflict free labels as a common objective. The computational times are not compared since the computational tests were realized in different machines.

Now, to assess the quality provided by LagClus when there are cartographic preferences assigned to a candidate position, we elaborated another experiment with MNCP2 formulation but considering that $w_{i,j}=j$ $"i=1\ldots N$ and $"j=1\ldots 4$. Thus, in this case there are some positions that are prioritized. The average results are reported in Table 8. The columns are the same shown at Table 6 with an additional column showing the size of the problem. Note that we have reduced gaps except for instances with 25 points. Moreover, the proportion of free labels is reduced compared to results of Tables 5 and 6, mainly due to the penalties assigned to candidate positions.

6. Conclusion

This paper presented two new mathematical formulations for Point-feature Cartographic Label Placement problem aiming a better map legibility. The objective is to minimize the number of existing overlaps in a labeling of all points on a map. Based on these formulations we proposed a Lagrangean relaxation with initial partition in clusters. For many instances the results found are very close to the optimal solutions and better than those reported in the literature.

We believe that this work contributes for cartographic point labeling problems and can insight solutions to other related problems that can be formulated on conflict graphs.

Acknowledgements

The authors acknowledge the useful comments and suggestions of two anonymous referees and Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (proc. 141470/2003-8, 304598/2003-8 and 471480/2003-7) for partial financial support.

References

[1] Beasley, J. An Exact Two-Dimensional Non Guillotine Cutting Tree Search Procedure. Operations Research 1985; 33: 49-64.

[2] Björkund P., Värbrand P., Yuan D. Optimized Planning of Frequency Hopping in Cellular Networks. Computers & Operations Research 2005; 32: 169-186.

[3] Chaudry S., McCormick S., Moon I. D. Locating Independent Facilities with Maximum Weight: Greedy Heuristics. Omega 1986; 14: 383-389.

[4] Christensen J., Marks J., Shieber S. Placing Text Labels on Maps and Diagrams. In: Heckbert P. (Ed.). Graphics Gems IV. Cambridge: Academic Press, 1994.

[5] Christensen J., Marks J., Shieber S. An Empirical Study of Algorithms for Point-Feature Label Placement. ACM Transactions on Graphics 1995; 14(3): 203-232.

[6] Church R. L., Murray A. T., Weintraub, A., Locational Issues in Forest Management. Location Science 1998; 6:137-153.

[7] Dowsland K. A. An Exact Algorithm for the Pallet Loading Problem. European Journal of Operational Research 1987; 31: 78-84.

[8] Fisher M. L. An Applications Oriented Guide to Lagrangian Relaxation. Interfaces 1985; 15: 10-21.

[9] Gerrard R. A., Church, R. L. Closest Assignment Constraints and Location Models: Properties and Structure. Location Science 1996; 4: 251-270.

[10] Goycoolea M., Murray A. T., Barahona F., Epstein R., Weintraub A. Harvest Scheduling Subject to Maximum Area Restrictions: Exploring Exact Approaches. Operations Research 2005, (to appear).

[11] Held M., Karp R. M. The Traveling Salesman Problem and Minimum Spanning Trees: Part II. Mathematical Programming 1971; 1: 6-25.

[12] Hicks I. V., Warren J. S., Warrier D., Wilhelm W. E. A Branch-and-Price Approach for the Maximum Weight Independent Set Problem. Working paper. (On line) http://ie.tamu.edu/People/faculty/Hicks/, 2004.

[13] Hirsch S. A. An Algorithm for Automatic name Placement around Point Data. American Cartographer 1982; 9(1): 5-17.

[14] ILOG. Cplex 7.5, Reference Manual. Mountain View, CA, 2001.

[15] Joseph D., Meidanis J., Tiwari, P. Determining DNA Sequence Similarity Using Maximal Independent Set Algorithms for Interval Graphs. In: Nurmi O, Ukkonen E (Eds). Algorithm Theory. SWAT 1992.

[16] Karypis G., Kumar V. Multilevel *k-way* Partitioning Scheme for Irregular Graphs. Journal of Parallel and Distributed Computing 1998; 48: 96-129.

[17] Lorena L. A. N., Furtado, J. C. Constructive Genetic Algorithm for Clustering Problems. Evolutionary Computation 2001; 9(3): 309-327.

[18] Moon I. D., Chaudhry S. An Analysis of Network Location Problems with Distance Constraints. Management Science 1984; 30: 290-307.

[19] Murray A. T., Church R. L. Solving the Anti-Covering Location Problem Using Lagrangian Relaxation. Computers & Operations Research 1996; 24(2): 127-140.

[20] Murray A. T., Church R. L. Constructing and Selecting Adjacency Constraints. INFOR 1996; 232-248.

[21] Murray A. T., Church R. L. Facets for Node Packing. European Journal of Operational Research 1996; 101: 598-608.

[22] Narciso M. G., Lorena, L. A. N. Lagrangean/Surrogate Relaxation for Generalized Assignment Problems. European Journal of Operational Research 1999; 114: 165-177.

[23] Parker R. G., Rardin R. L. Discrete Optimization. Academic Press, New York, USA, 1988.

[24] Schreyer M., Raidl G. R. Letting Ants Labeling Point Features. In Proc. of the 2002 IEEE Congress on Evolutionary Computation at the IEEE World Congress on Computational Intelligence; 2002, 1564-1569.

[25] Strijk T., Verweij B., Aardal K. Algorithms for Maximum Independent Set Applied to Map Labeling. (On line), ftp://ftp.cs.uu.nl/pub/RUU/CStechreps/CS-2000/2000-22.ps.gz., 2000.

[26] Verner O. V., Wainwright R. L., Schoenefeld D. A. Placing Text Labels on Maps and Diagrams Using Genetic Algorithms with Masking. INFORMS Journal on Computing 1997; 9: 266-275.

[27] Yamamoto M., Camara G., Lorena L.A.N. Tabu Search Heuristic for Point-Feature Cartographic Label Placement. GeoInformatica and International Jour nal on Advances of Computer Science for Geographic Information Systems. Kluwer Academic Publisher. Netherlands, 2002, 6(1): 77-90.

[28] Yamamoto M., Lorena L.A.N. A Constructive Genetic Approach to Point-Feature Cartographic Label Placement. In: Ibaraki, T., No nobe, K. and Yagiura, M. (Eds.). Metaheuristics: Progress as Real Problem Solvers. Kluwer Academic Publishers, 2005. p. 285-300.

[29] Yamamoto M. Novos Algoritmos para Rotulação Cartográfica de Pontos (New Algorithms for Point-Feature Cartographic Label Placement). PhD Dissertation on Applied Computation, INPE- Brazilian Space Research Institute, Brazil, 2003.

[30] Zoraster S. Integer Programming Applied to the Map Label Placement Problem. Cartographica 1986; 23(3): 16-27.

[31] Zoraster S. The Solution of Large 0-1 Integer Programming Problems Encountered in Automated Cartography. Operations Research 1990; 38(5): 752-759.

[32] Zoraster S. Expert Systems and the Map Label Placement Problem. Cartographica 1991; 28(1): 1-9.

[33] Zoraster S. Practical results using simulated annealing for point feature label placement. Cartography and GIS, 1997; 24(4): 228-238.

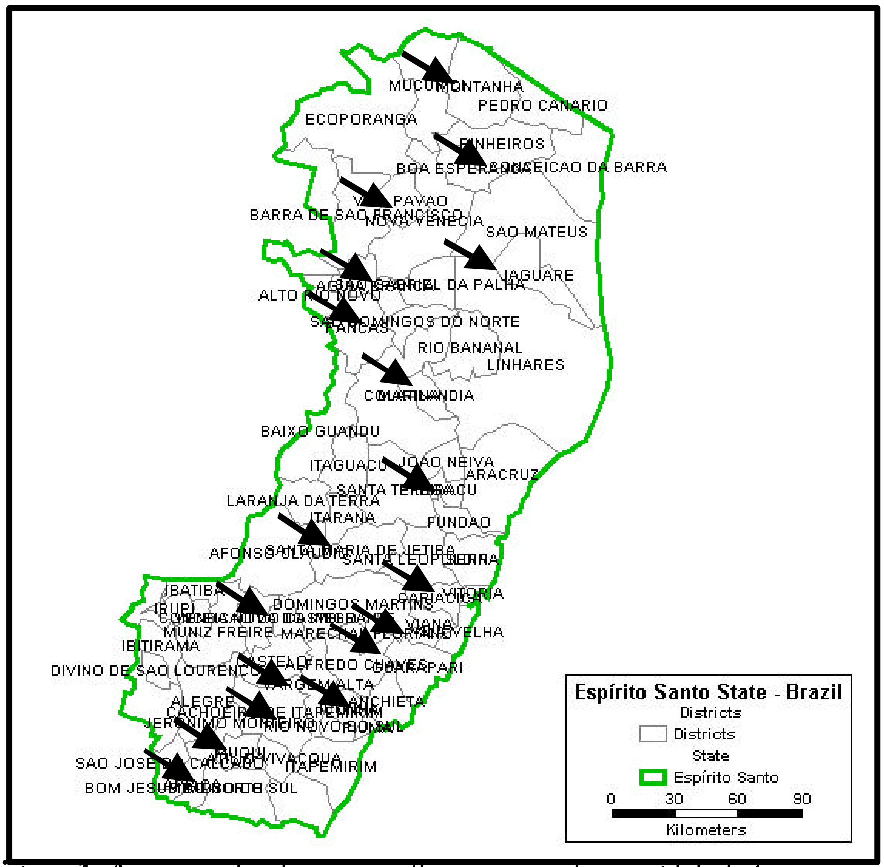

Figure 1. An example of a map with some overlapping labels (see arrows)

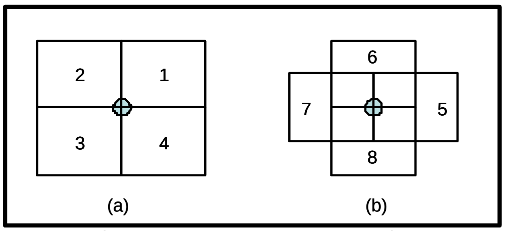

Figure 2. Set of 8 candidate positions for one point [5].

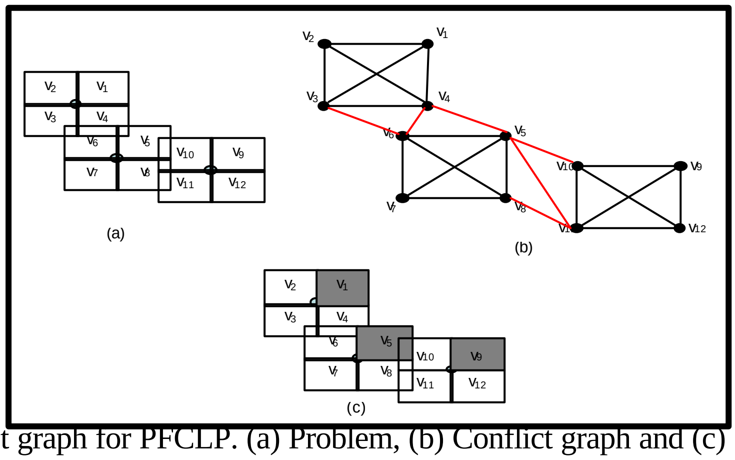

Figure 3. Conflict graph for PFCLP. (a) Problem, (b) Conflict graph and (c) Optimal solution.

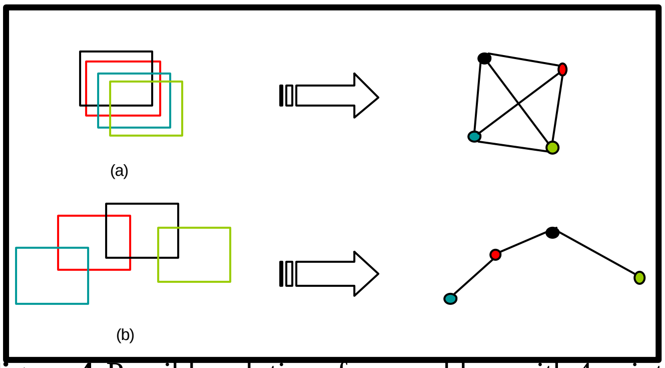

Figure 4. Possible solutions for a problem with 4 points.

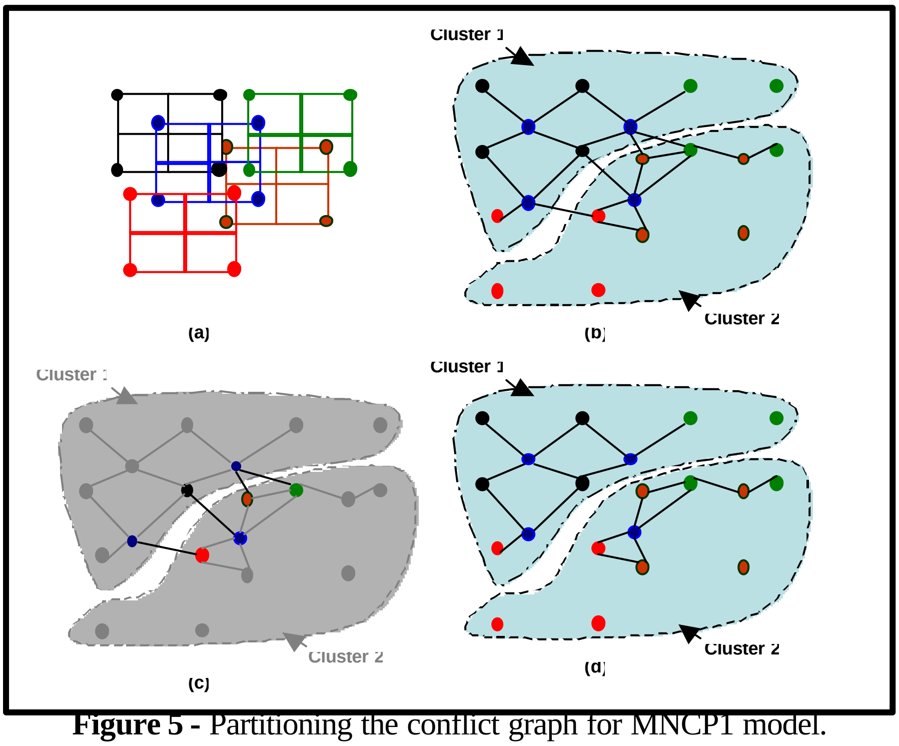

Figure 5 - Partitioning the conflict graph for MNCP1 model.

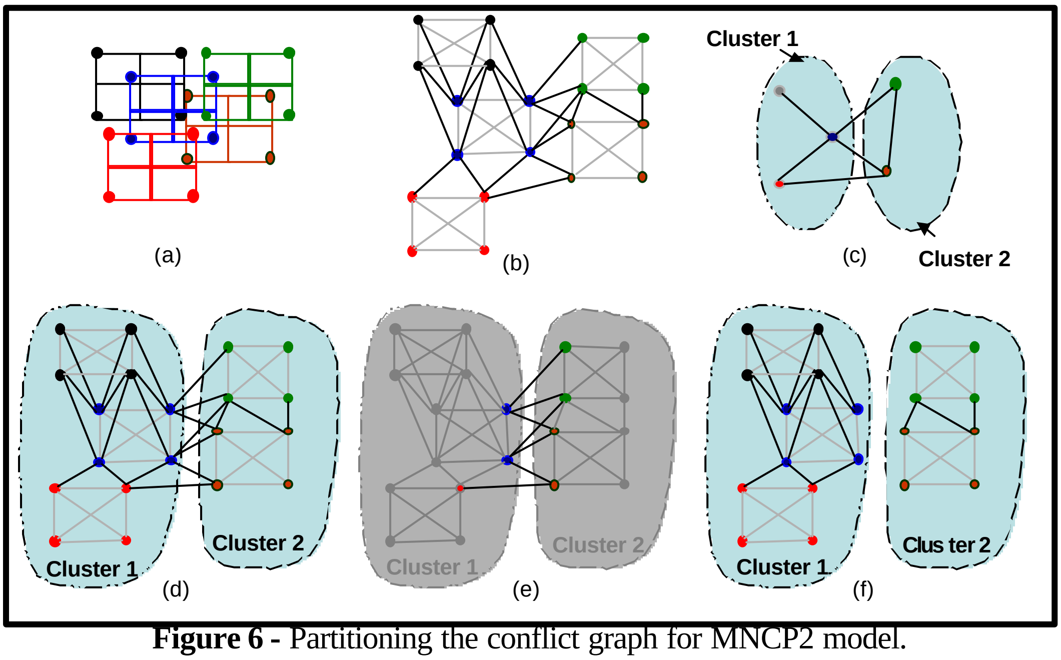

Figure 6 - Partitioning the conflict graph for MNCP2 model.

Table 1. Average number of constraints generated by MNCP1 and MNCP2

Number of points	Number of instances	MNCP1	MNCP2
25	8	357	96
100	25	202	153
250	25	864	530
500	25	2909	1412
750	25	6181	2481
1000	25	10700	3643

Table 2. Average results for MNCP1 model with CPLEX [14].

MNCP1						
Problem	**Lower bound**	**Upper bound**	**GAP**	**Time (s)**	**Labels in conflict**	**Proportion of free labels**
25	27.75	27.75	0.00%	1.60	4.88	100.00%
100	100.00	100.00	0.00%	0.02	0.00	100.00%
250	250.00	250.00	0.00%	0.06	0.00	100.00%
500	500.84	500.84	0.00%	3.12	1.68	99.67%
750	756.13	759.36	0.42%	6586.88	18.40	97.55%
1000	1005.76	1048.88	4.11%	5258.98	90.76	90.92%

Table 3. Average results for MNCP2 model with CPLEX [14].

MNCP2						
Problem	**Lower bound**	**Upper bound**	**GAP**	**Time (s)**	**Labels in conflict**	**Proportion of free labels**
25	27.75	27.75	0.00%	0.20	4.88	80.50%
100	100.00	100.00	0.00%	0.03	0.00	100.00%
250	250.00	250.00	0.00%	0.06	0.00	100.00%
500	500.84	500.84	0.00%	0.74	1.68	99.67%
750	757.25	759.12	0.25%	9625.92	17.84	97.62%
1000	1010.37	1051.92	3.94%	6683.80	97.12	90.29%

Table 4. Average results with LagClus for MNCP1

Problem	Number of Clusters	Lower bound	Upper bound	GAP	Time (s)	Labels in conflict	Proportion of free labels
25	2	25.13	27.88	9.69%	23.88	5.63	77.50%
100	4	100.00	100.00	0.00%	0.16	0.00	100.00%
250	10	250.00	250.00	0.00%	2.36	0.00	100.00%
500	20	498.43	501.52	0.61%	82.72	3.08	99.38%
750	25	749.41	767.08	2.30%	337.80	33.56	95.53%
1000	60	1002.11	1070.60	6.39%	817.00	135.32	86.47%

Table 5. Average results with LagClus for MNCP2

Problem	Number of Clusters	Lower bound	Upper bound	GAP	Time (s)	Labels in conflict	Proportion of free labels
25	2	25.62	28.13	8.67%	3.50	6.00	76.00%
100	2	100.00	100.00	0.00%	0.12	0.00	100.00%
250	2	250.00	250.00	0.00%	0.12	0.00	100.00%
500	2	500.84	500.84	0.00%	0.40	1.68	99.67%
750	10	758.09	758.96	0.12%	53.84	17.60	97.65%
1000	25	1030.07	1047.32	1.64%	3445.40	90.16	90.98%

Table 6. Average results obtained with LagClus for MNCP2 on problems with 1000 points.

Number of Clusters	Lower Bound	Upper Bound	GAP	Time (s)	Labels in conflict	Proportion of free labels
20	1031.23	1044.80	1.30%	3842.84	85.80	91.42%
25	1030.07	1047.32	1.64%	3445.40	90.16	90.98%
30	1026.81	1049.16	2.13%	734.80	93.56	90.64%

Table 7. Comparison with the literature

Algorithm	Proportion of free labels				
	Problems				
	100	250	500	750	1000
LagClus	100.00	100.00	99.67	97.65	91.42/90.98/90.64
PMNC Exact – CPLEX	100.00	100.00	99.67	97.62	90.92
CGA_{Best} [28]	100.00	100.00	99.60	97.10	90.70
$\text{CGA}_{\text{Average}}$ [28]	100.00	100.00	99.60	96.80	90.40
Tabu Search [27]	100.00	100.00	99.30	96.80	90.00
GA with masking [26]	100.00	99.98	98.79	95.99	88.96
GA [26]	100.00	98.40	92.59	82.38	65.70
Simulated Annealing [5]	100.00	99.90	98.30	92.30	82.09
Zoraster [31]	100.00	99.79	96.21	79.78	53.06
Hirsh [13]	100.00	99.58	95.70	82.04	60.24
3-opt Gradient Descent [5]	100.00	99.76	97.34	89.44	77.83
2-opt Gradient Descent [5]	100.00	99.36	95.62	85.60	73.37
Gradient Descent [5]	98.64	95.47	86.46	72.40	58.29
Greedy Algorithm [5]	95.12	88.82	75.15	58.57	43.41

Table 8. Average results obtained with LagClus for MNCP2 considering penalty for the candidate positions.

Problem	Number of Clusters	Lower bound	Upper bound	GAP	Time(s)	Labels in conflict	Proportion of free labels
25	2	38.217	43.750	12.65%	2.750	18.000	28.00%
100	2	106.60	106.60	0.00%	0.00	9.12	90.88%
250	2	286.80	286.80	0.00%	0.16	52.00	79.20%
500	2	638.80	638.80	0.00%	0.24	163.00	67.40%
750	10	1049.47	1050.68	0.12%	30.36	335.56	55.26%
1000	20	1509.32	1528.08	1.23%	248.28	545.32	45.47%
1000	25	1504.27	1529.24	1.63%	267.96	549.44	45.06%
1000	30	1499.16	1530.40	2.04%	183.32	552.66	44.73%