versão impressa ISSN 0101-7438 / versão online ISSN 1678-5142

LAGRANGEAN RELAXATION BOUNDS FOR POINT-FEATURE CARTOGRAPHIC LABEL PLACEMENT PROBLEM

Glaydston Mattos Ribeiro
Computer and Applied Mathematics Laboratory (LAC)
Brazilian Space Research Institute (INPE)
São José dos Campos – SP
glaydston@lac.inpe.br

Luiz Antonio Nogueira Lorena *
Computer and Applied Mathematics Laboratory (LAC)
Brazilian Space Research Institute (INPE)
São José dos Campos – SP
lorena@lac.inpe.br

* *Corresponding author* / autor para quem as correspondências devem ser encaminhadas

Recebido em 12/2004; aceito em 07/2006 após 1 revisão
Received December 2004; accepted July 2006 after one revision

Abstract

The objective of the point-feature cartographic label placement problem (PFCLP) is to give more legibility to an automatic map creation, placing point labels in clear positions. Many researchers consider distinct approaches for PFCLP, such as to obtain the maximum number of labeled points that can be placed without overlapping or to obtain the maximum number of labeled points without overlaps considering that all points must be labeled. This paper considers another variant of the problem in which one has to minimize the number of overlaps while all points are labeled in the map. A conflict graph is initially defined and a mathematical formulation of binary integer linear programming is presented. Commercial optimization packages could not solve large instances exactly using this formulation over instances proposed in the literature. A heuristic is then examined considering a Lagrangean relaxation performed after an initial partition of the conflict graph into clusters. This decomposition allowed us to introduce tight lower and upper bounds for PFCLP.

Keywords: label placement; modeling; Lagrangean relaxation.

Resumo

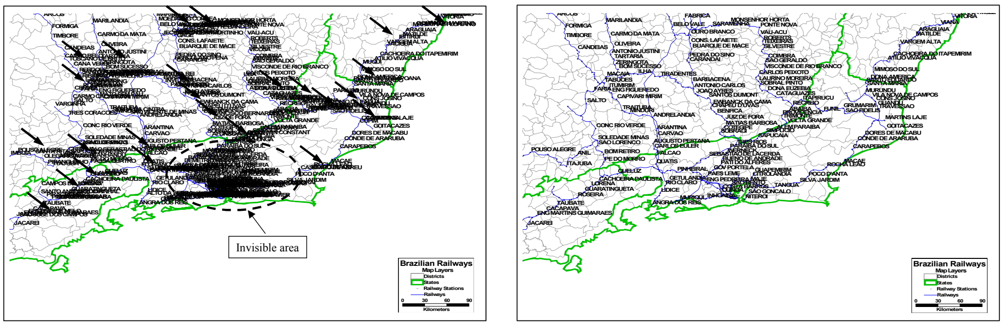
O Problema Rotulação Cartográfica de Pontos (PRCP) tem como objetivo dar maior legibilidade a um mapa, colocando os rótulos dos pontos em posições legíveis. Existem abordagens distintas para o PRCP direcionadas a obter o máximo número de pontos rotulados que podem ser colocados sem sobreposição ou ainda obter o máximo número de pontos rotulados sem sobreposição considerando que todos os pontos devem ser rotulados. Esse artigo aborda o problema de uma outra forma, minimizando o número de sobreposições existentes em uma rotulação de todos os pontos. Um grafo de conflitos é definido inicialmente e uma formulação matemática de programação linear inteira binária é apresentada. Instâncias de grande porte propostas na literatura não puderam ser resolvidas por um *software* comercial de otimização, com isso, uma heurística é examinada considerando uma relaxação Lagrangeana feita após um particionamento inicial do grafo de conflitos em *clusters*. Essa decomposição permitiu obter bons limitantes inferiores e superiores para o PRCP.

Palavras-chave: rotulação de pontos; relaxação Lagrangeana; modelagem.

1. Introduction

The point-feature cartographic label placement problem (PFCLP) is a challenge problem in automated cartography. Positioning the texts requires that overlaps among texts should be avoided and also cartographic conventions and preferences should be obeyed. Figure 1(a) illustrates the difficulty that arises when many labels are positioned in overlapping positions generating invisible areas (see arrows). Figure 1(b) shows a solution for this problem where some points are not labeled.

Although it can be better to label only some points to produce a cleaner map, there are some geographic applications where all points must be labeled. Thus, we need approaches and algorithms to generate the best possible maps.



(a)

(b)

Figure 1 – Map of Brazilian Railway Stations. An example of a map with some overlapping labels.

PFCLP seeks to place point labels in positions in such a way that a set of constraints are satisfied, minimizing or maximizing an objective function. However, a list of candidate positions is presented for each point, indicating where a label can be placed. The list is chosen in accordance to cartographic standards (Christensen *et al.*, 1995) that prioritize certain positions. Figure 2(a) shows a group of 8 candidate positions for a point, where the numbers indicate the cartographic preference in an increasing order.

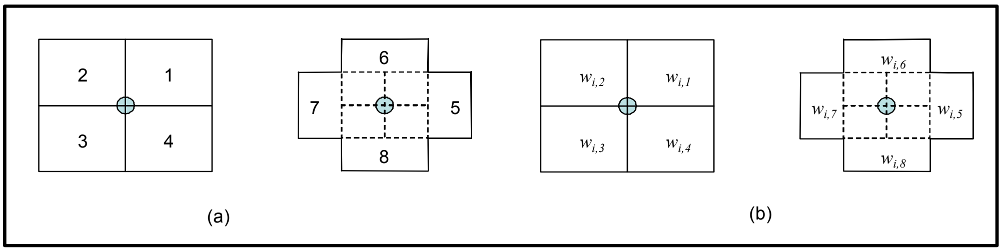


Figure 2 – Set of 8 candidate positions for one point (Christensen *et al.*, 1995).

Placing labels in candidate positions can generate overlaps (conflicts) compromising the map visibility. Thus, due to these potential overlaps, the PFCLP with *N* points can be represented through a graph $G=\{V,E\}$, where *V* is a set of the candidate positions (vertices) and *E* a set of edges representing overlaps or conflicts. Figure 3(b) shows the conflict graph of the example shown in Figure 3(a). This example has three points, each one with 4 candidate positions. The candidate position v_3 has potential conflicts with positions v_1, v_2, v_4 and v_6, v_4 has potential conflicts with v_1, v_2, v_3, v_5 and v_6, and so on. Figure 3(c) shows a solution composed by v_1, v_5 and v_9 that is optimal for this problem because it does not present conflicts between labels.

Starting from this conflict graph representation, two different approaches are usually considered for PFCLP. This problem can be considered as a Maximum Independent Vertex Set Problem (MIVSP) (Zoraster, 1990; Strijk *et al.*, 2000) or as a Maximum Number of Conflict Free Labels Problem (MNCFLP) (Christensen *et al.*, 1995). In both problems, the optimal value refers to the number of points in the final solution whose labels are not conflicting. However, the constraints requiring the labeling of a point are treated differently.

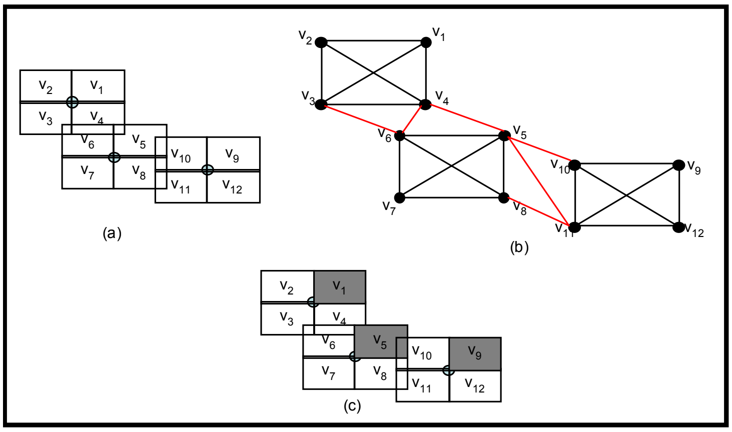


Figure 3 – Candidate positions (a), conflicts graph (b) and optimal solution (c).

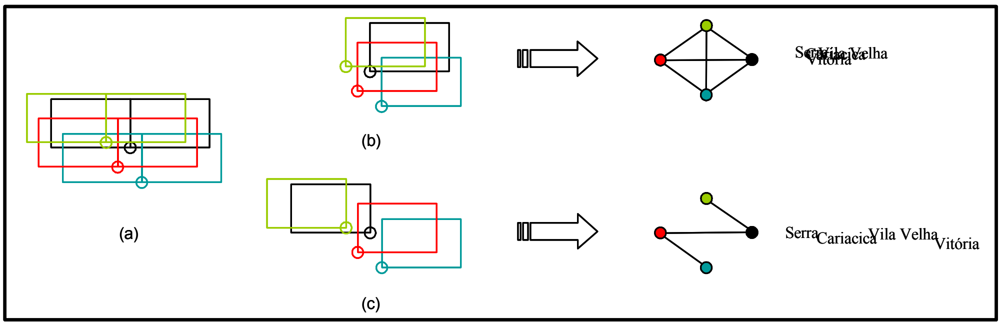


Figure 4 – A labeling problem with 4 points and 2 candidate position for each one (a). Two possible solutions are shown at (b) and (c).

The MNCFLP is more useful under the cartographic point of view than the MIVSP but the map visibility is not fully explored (Ribeiro, 2005). Figure 4 shows two possible solutions for the same instance. The cost of the both solutions is the same for MNCFLP while, if we count the number of conflicts (edges) in their graphs, they differ by three units. At right of this figure we have an example that shows the difference between these solutions.

Considering the map visibility above and that some cartographic problems need to label all points in maps, diagrams or graphs, this paper proposes a variant of the problem in which one has to minimize the number of overlaps while all points are labeled in a map. We propose for this variant an integer linear programming model and present a Lagrangean heuristic that is based on a decomposition of the conflict graph. Computational experiments showed that for instances up to 500 points, this new model can be easily computed by a commercial solver. Consequently, for such situations, these packages can be used in cartographic centers.

The rest of the paper is described as follows. Next section presents a brief review about PFCLP, followed by Section 3 that presents the mathematical model proposed. Section 4 presents the Lagrangean relaxation proposed and some heuristics that are used in the subgradient algorithm. Section 5 shows the computational results using instances formed by standard sets of randomly generated points suggested in the literature, and Section 6 has our conclusions and suggestions for further research.

2. Literature Review

The Maximal Independent Vertex Set Problem (MIVSP) presents a substantial research considering algorithms and heuristics. Besides, the MIVSP has several applications in different fields such as in DNA sequencing (Joseph *et al.*, 1992), location of military defenses (Chaudhry *et al.*, 1986), location-allocation models (Gerrard & Church, 1996), anti-covering location (Murray & Church, 1996a), forest planning (Murray & Church, 1996b) and Church *et al.* (1998), cut and packing (Beasley, 1985) and pallet loading (Dowsland, 1987).

Specifically considering the MIVSP as a PFCLP, Zoraster (1986, 1990 and 1991) formulated mathematically the PFCLP working with conflict constraints and dummy candidate positions of high cost if the points 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.* (2000) proposed new mathematical formulations and examined a Tabu Search algorithm, obtaining interesting results for their instances. The authors explored some kind of constraints that are known as cut constraints, presented previously by Murray & Church (1996c) and Moon & Chaudhry (1984).

The Maximum Number of Conflict Free Labels Problem (MNCFLP) was examined in several works. Christensen *et al.* (1993; 1995) proposed an Exhaustive Search Approach, alternating positions of the labels that were previously positioned. Christensen *et al.* (1995) also proposed a Greedy Algorithm and a Discrete Gradient Descent Algorithm. These algorithms have difficulty to escape from local maxima. Hirsh (1982) developed a Dynamic Algorithm of label repulsion, where labels in conflicts are moved trying to remove a conflict. Verner *et al.* (1997) 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.* (2002) proposed a Tabu Search algorithm for the MNCFLP that provides good results compared to other methods from the literature. Yamamoto & Lorena (2003) developed an exact algorithm for small instances of PFCLP and applied the Constructive

Genetic Algorithm (CGA) proposed by Lorena & Furtado (2001) 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. However, the authors can not prove the optimality of these results because CGA is a metaheuristic.

Although the MNCFLP presents several different algorithms, it has not a mathematical formulation like the model proposed by Zoraster (1991). However, almost all heuristics proposed for solving the MNCFLP uses the conflict graph as a base for their mechanism.

Thus, considering these characteristics, in the next section we propose a new mathematical model that combines the conflict graph and the Zoraster's formulation for constructing a new approach that minimizes the number of conflicts (edges) in a conflict graph. This new mathematical model allows us to label all points.

3. Mathematical Formulation

This section presents a new approach and a mathematical formulation for the PCFLP that minimizes the number of conflicts. This approach can be used in problems where we have to label all points.

Considering that each point *i* has a set P_i of candidate positions, as shown at Figure 2(a), we start by defining the variables used in the model. So, each candidate position is represented by a binary variable $x_{i,j}$ where $i \in \{1,...,N\}$, $j \in \{1,...,P_i\}$, and *N* is the number of points that will be labeled. If $x_{i,j} = 1$ the candidate position *j* for 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 the point *i*, a cost (a penalty) $w_{i,j}$ is assigned. It represents the cartographic preferences as shown at Figure 2(b).

For each candidate position $x_{i,j}$ there is a set $S_{i,j}$ of index pairs (*k,t*) that corresponds to candidate positions $x_{k,t}$ that present potential conflicts with $x_{i,j}$. For all $(k,t) \in S_{i,j}$, where $k \in \{1,...,N\} : k > i$ and $t \in \{1,...,P_k\}$, there is 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 (MNCP) for the PFCLP can be represented by:

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

For each point *i* exactly one of its candidate positions must be chosen. This set of constraints can be written as:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \qquad \forall\ i = 1...N \qquad (2)$$

We also must take into account that when a label is chosen, it can be overlapping the other ones. So, a new set of constraints is necessary:

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

Thus, the MNCP can be formulated as a binary integer linear programming problem:

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

$$\textit{Subject to} \quad \sum_{j=1}^{P_i} x_{i,j} = 1 \qquad \forall\, i = 1...N \tag{5}$$

$$x_{i,j} + x_{k,t} - y_{i,j,k,t} \le 1 \qquad \forall\, i = 1...N \quad \forall\, j = 1...P_i \quad (k,t) \in S_{i,j} \tag{6}$$

$$x_{i,j}, x_{k,t} \;\textit{and}\; y_{i,j,k,t} \in \{0,1\} \qquad \forall\, i = 1...N \quad \forall\, j = 1...P_i \quad (k,t) \in S_{i,j} \tag{7}$$

Constraint (7) ensures that all decision variables of the problem are binaries. Depending on the values assigned for the costs $w_{i,j}$, the conflict variables y can be reduced to zero and a cleaner map is obtained.

We tested the formulation above with CPLEX 7.5 (ILOG, 2001) on a set of standard problems with four candidate positions for each point, proposed by Yamamoto & Lorena (2003). The optimal solution could be found in few seconds for the instances up to 500 points. For larger instances (750 and 1000 points), CPLEX could not obtain the optimal solutions in few hours reaching an out of memory state in a computer with Pentium IV 2.66 GHz processor and 512 MB of RAM memory.

Thus, to provide bounds for the MNCP, we show in next section a Lagrangean relaxation that differs from the literature. It works with several sub-problems (clusters) that are generated by partitioning of the conflict graph.

4. Lagrangean Relaxation With Clusters

Prior to explain the Lagrangean relaxation, we start by observing that the conflict graph generated by PFCLP, provides clusters of candidate positions. For example, Figure 5 shows a conflict graph generated by a problem with 250 points where each one has four candidate positions. The black vertices represent the maximum independent set (Strijk *et al.*, 2000). It is easy to see that this graph is sparse and presents well-defined clusters of candidate positions (see stippled lines). So, if we relax the edges that are connecting the clusters, raise sub-problems (clusters) that can be solved independently.

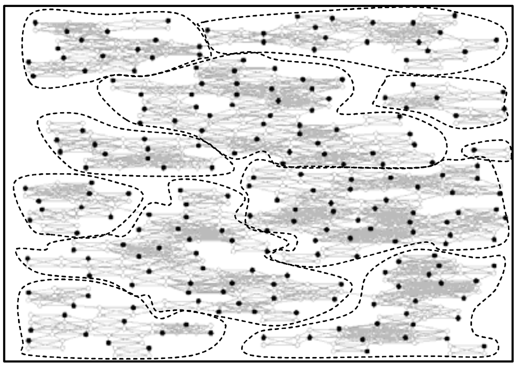


Figure 5 – Clusters provided by a conflict graph of a map labeling problem on 250 points. (Strijk *et al.*, 2000)

Thus, considering this PFCLP characteristic, the Lagrangean relaxation proposed here has two distinct phases. The first one divides the conflict graph into clusters and in the second one, the edges connecting the clusters are relaxed in a Lagrangean fashion. Depending on the size of these sub-problems, each one sub-problem can be hard to solve and thereby the number of clusters is essential to obtain good bounds in a reasonable time.

Figure 6 details the Lagrangean relaxation. The graph obtained for the problem shown in Figure 6(a) is partitioned into two clusters (b). Note that for each point, the clique generated by their candidate positions is initially ignored. In this partition some constraints represented by edges inter clusters are removed (c) and two small problems (d) can be independently solved. Thus, the edges inter clusters and those that compose the clique for each point, are relaxed in a Lagrangean fashion.

So, after this decomposition, each cluster has only conflict constraints (6) and even so, the clusters can be hard problems to be solved. These sub-problems are solved in a subgradient algorithm, providing bounds for the problem. Depending on the number of clusters considered this Lagrangean relaxation can be more or less stronger.

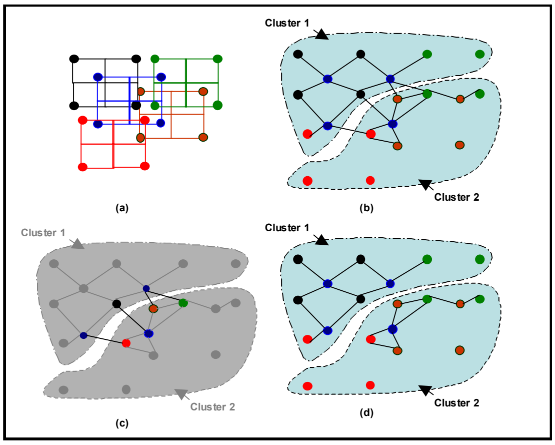


Figure 6 – Partitioning the conflict graph. Problem (a), conflict graph and the clusters (b), (c) the edges inter clusters; and (d) two sub-problems.

Therefore, considering the original problem *P*, this Lagrangean relaxation must be applied following these steps:

i. Apply a graph partitioning heuristic to divide *G* into *m* clusters. The problem *P* can be written through the objective function defined in (4) subject to (5), (6) and (7), where, the conflict constraints (6) is now divided into 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. Using two distinct multipliers, relax in a Lagrangean way, the constraints (5) and the conflict constraints corresponding to edges connecting the clusters.

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

Relaxing constraints (5), the relaxed solution cannot be feasible to *P* because it is possible that for a point, no one candidate position is assigned or even more than one. So, the following heuristics CH and IH are used to obtain and improve a feasible solution.

Constructive Heuristic – CH

Let:

- *Sol_point_i* be a set of candidate positions of the point *i.*

```
1. FOR i=1 to N DO
   2. Sol_point_i ← Find in relaxed solution all candidate positions different from zero for
      the point i.
   3. IF | Sol_point_i | <> 0 THEN
         4. Select for feasible solution the candidate position $j \in$ Sol_point_i that
            presents the smallest number of conflicts with the current feasible solution.
            In case of tie, select the candidate position with the smallest cardinality set
            $S_{i,j}$.
   ELSE
         5. Select the candidate position $j \in P_i$ with the smallest cardinality set $S_{i,j}$.
   END FOR
```

Improvement Heuristic – IH

Let:

- *Conflict_Array* be an array with *N* positions that stores the number of overlaps for each point *i* related to the current feasible solution;
- *Curr_Feasible_Solution* be an array with the current feasible solution;
- *Best_Candidate_i* be an integer variable that stores the best candidate position of the point *i* to enter in the *Curr_Feasible_Solution*.

```
1. Compute Conflict_Array.
2. FOR i=1 to N DO
   3. IF Conflict_Array[ i ]<>0 THEN
         4. Best_Candidate_i ← Select the candidate position $j \in P_i$ that presents the
            smallest number of conflicts with the current solution
            Curr_Feasible_Solution.
         5. Update Curr_Feasible_Solution with Best_Candidate_i.
         6. Compute Conflict_Array.
   END FOR
```

For the computational experiments, the sub-problems were solved by CPLEX in reasonable times. The number of clusters was defined experimentally. The partitioning of graph G was obtained using METIS (Karyps & Kumar, 1998), a well-known heuristic for Graph Partitioning Problems. Given a conflict graph G and a pre-defined number m of clusters, METIS divides the graph into m clusters minimizing the number of edges with extremities in different clusters. Recently Hicks *et al.* (2004) found good results applying this technique in a Branch-and-Price algorithm to Maximum Weight Independent Set Problems.

A subgradient algorithm is used for solving the Lagrangean dual (Parker & Rardin, 1988). The subgradient method is similar to the one proposed by Held & Karp (1971) and updates the multipliers considering step sizes based on the relaxed solutions and the feasible solutions obtained by heuristics CH and IH. We implemented the subgradient algorithm described by Narciso & Lorena (1999) and the stopping tests used were: step size less or equal than 0.005, difference between upper and lower bounds less or equal than 1 or subgradient norm equals to 0.

5. Computational Results

The computational tests are performed over standard sets of randomly generated points proposed by Yamamoto & Lorena (2003), available at http://www.lac.inpe.br/~lorena/instancias.html. These sets are composed of twenty five instances for each number of points $N \in \{25, 100, 250, 500, 750, 1000\}$. We considered as Zoraster (1990), Yamamoto *et al.* (2002) and Yamamoto & Lorena (2003), cost or penalty equal to 1 for all the candidate positions, being the number of those positions equal to 4. Observe that the particular case of $w_{i,j}=1$ in (4)-(7) has a trivial lower bound equal to N, when all points are labeled without conflicts.

We implemented the subgradient algorithm in C++ and the tests were performed in a computer with Pentium IV 2.66 GHz processor and 512 MB of RAM memory.

Table 1 shows the clustering information for each instance class. The first column presents the number of points, followed by the number of clusters considered and the number of possible vertices in each cluster. Those numbers of clusters reported at Table 1 were defined based on prior experiments of the authors, see Ribeiro (2005).

Table 1 – Cluster information for each instance class.

Instance	Number of Clusters	Number of Possible Positions in each Cluster
25	2	50
100	4	100
250	10	100
500	20	100
750	25	120
1000	60	~67

Table 2 reports the *LagClus* average results over twenty five instances for each number of points. We used CPLEX 7.5 for solving the binary integer linear sub-problems.

The information in columns are: *Problem* – Number of points to be labeled; *Optimal Solution* and *Time$_1$ (s)* – The optimal solution and time elapsed by CPLEX applied over formulation (4) – (7); *Lower bound* – The best dual limit found by relaxation; *Upper bound* – The best upper bound (feasible solution) found by heuristics CH and IH; *Gap_ub* – Percentage deviation from optimal solution to the best upper bound: $Gap\_ub = \left(\frac{Upper\ bound - Solution}{Solution}\right)*100$; *Gap_lb* – Percentage deviation from optimal solution to the best lower bound: $Gap\_lb = \left(\frac{Solution - Lower\ bound}{Solution}\right)*100$; *Iter* – Number of the iterations used by subgradient algorithm and *Time$_2$ (s)* – The total computational time elapsed by subgradient algorithm reaching some stop condition.

The computational results for *LagClus* shown in Table 2 are very promising. The large problems (750 and 1000 points) were solved, in average, in 337,80 and 817,00 seconds, respectively. The upper bound gaps varied from 0,00% to 0,46%, very close to the optimal solution. The lower bound gaps varied from 0,00% to 9,27% and some of them improved the trivial number of points limit (see 25 and 1000 points). The problems with 100 and 250 points are simple and solutions without conflicts are obtained quickly. In these cases, the subgradient algorithm stops and we considered that the lower bounds are equal to the upper bounds.

Table 2 – Average results for *LagClus* – Bounds, Gaps, Iterations and Computational Times.

Problem	CPLEX 7.5		LAGCLUS					
	Optimal Solution	Time$_1$ (s)	Lower bound	Upper bound	Gap UB (%)	Gap LB (%)	Iter	Time$_2$ (s)
25	27.75	1.60	25.13	27.88	0.46	9.27	148.50	23.88
100	100.00	0.02	100.00	100.00	0.00	0.00	1.00	0.16
250	250.00	0.06	250.00	250.00	0.00	0.00	7.12	2.36
500	500.84	3.12	498.43	501.52	0.14	0.48	103.16	82.72
750	-	-	749.41	767.08	-	-	145.28	337.80
1000	-	-	1002.11	1070.60	-	-	145.96	817.00

Table 3 – Average results for *LagClus* – Bounds, Iterations and Computational Times.

Problem	Number of clusters	LAGCLUS			
		Lower bound	Upper bound	Iter	Time (s)
750	20	749.63	767.28	145,48	436.04
750	25	749.41	767.08	145.28	337.80
750	30	749.57	767.32	145.40	123.12

To show what happens when we consider different number of clusters, we performed some experiments for instances with 750 points and Table 3 reports the average results provided by *LagClus*. Note that when the number of cluster increases, the computational time decreases although the bounds remain practically constant.

Table 4 shows the results obtained for a simple Lagrangean relaxation over the set of constraints (5). CPLEX 7.5 was also used for solving the Lagrangean relaxations. The computational times increased drastically. The upper bound was not improved and the large-scale instances (of 500, 750, and 1000 points) could not be solved. For example, the instance number 7 of problems with 25 points elapsed 444 seconds to be completed.

Table 4 – Average results for a Lagrangean relaxation of constraints (5).

Problem	**Lagrangean relaxation over constraint set (5)**					
	Lower bound	**Upper bound**	**Gap UB (%)**	**Gap LB (%)**	**Iter**	**Time (s)**
25	25.13	28.38	2.29	9.27	151.63	104.63
100	100.00	100.00	0.00	0.00	1.00	0.16
250	250.00	250.00	0.00	0.00	1.48	0.92

PS: The solutions were not obtained for problems with 500, 750 and 1000 points due to time-consuming conditions.

Relaxations of constraints (6) were also tested. The dual bounds were always smaller than the trivial limit imposed by the number of points (size) of the problem. So, relaxing the constraint set (6) could be considered weak for these instances. For more details, see Ribeiro (2005).

It is important to note that all approaches revised in section 2 have different objectives of that in MNCP and consequently the computational results are not comparable.

6. Conclusions

This paper presented a new approach and a new mathematical model for point-feature cartographic label placement problem aiming a better map legibility. This model seeks minimize the number of existing overlaps, labeling all points on a map.

A new relaxation heuristic was also proposed. This relaxation works with clusters and presented tight bounds on a set of instances varying from 25 up to 1000 points. For many instances the results found are very close to the optimal solutions.

We believe that this work contributes for point-feature cartographic label placement problems and the LagClus can be useful in related problems, for instance, the Maximal Vertex Independent Set Problem.

Acknowledgements

The authors acknowledge the useful comments and suggestions of three 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. (1985). An Exact Two-Dimensional Non Guillotine Cutting Tree Search Procedure. *Operations Research*, **33**, 49-64.

(2) Chaudhry, S.; McCormick, S. & Moon, I.D. (1986). Locating Independent Facilities with Maximum Weight: Greedy Heuristics. *Omega*, **14**, 383-389.

(3) Christensen, J.; Marks, J. & Shieber, S. (1993). *Graphics Gems IV*, chapter Placing Text Labels on Maps and Diagrams. London. Academic Press, Cambridge Mass.

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

(5) Church, R.L.; Murray, A.T. & Weintraub, A. (1998). Locational Issues in Forest Management. *Location Science*, **6**, 137-153.

(6) Dowsland, K.A. (1987). An Exact Algorithm for the Pallet Loading Problem. *European Journal of Operational Research*, **31**, 78-84.

(7) Gerrard, R.A. & Church, R.L. (1996). Closest Assignment Constraints and Location Models: Properties and Structure. *Location Science*, **4**, 251-270.

(8) Held, M. & Karp, R.M. (1971). The Traveling Salesman Problem and Minimum Spanning Trees: Part II. *Mathematical Programming*, **1**, 6-25.

(9) Hicks, I.V.; Warren, J.S.; Warrier, D. & Wilhelm W.E. (2004). A Branch-and-Price Approach for the Maximum Weight Independent Set Problem. <http://ie.tamu.edu/People/faculty/Hicks/>. Accessed March 15, 2004.

(10) Hirsch, S.A. (1982). An Algorithm for Automatic name Placement around Point Data. *American Cartographer*, **9**(1), 5-17.

(11) ILOG (2001). *Cplex 7.5 Reference Manual*. ©Copyright 2001 by Ilog.

(12) Joseph, D.; Meidanis, J. & Tiwari, P. (1992). Determining DNA Sequence Similarity Using Maximal Independent Set Algorithms for Interval Graphs. **In**: *Algorithm Theory – SWAT 92* [edited by O. Nurmi and E. Ukkonen], 326-337.

(13) Karypis, G. & Kumar, V. (1998). Multilevel *k-way* Partitioning Scheme for Irregular Graphs. *Journal of Parallel and Distributed Computing*, **48**, 96-129.

(14) Lorena, L.A.N. & Furtado, J.C. (2001). Constructive Genetic Algorithm for Clustering Problems. *Evolutionary Computation*, **9**(3), 309-327.

(15) Moon, I.D. & Chaudhry, S. (1984). An Analysis of Network Location Problems with Distance Constraints. *Management Science*, **30**, 290-307.

(16) Murray, A.T. & Church, R.L. (1996a). Solving the Anti-Covering Location Problem Using Lagrangian Relaxation. *Computers and Operations Research*, **24**(2), 127-140.

(17) Murray, A.T. & Church, R.L. (1996b). Constructing and Selecting Adjacency Constraints. *INFOR*, **34**, 232-248.

(18) Murray, A.T. & Church, R.L. (1996c). Facets for Node Packing. *European Journal of Operational Research*, **101**, 598-608.

(19) Narciso, M.G. & Lorena, L.A.N. (1999). Lagrangean/Surrogate Relaxation for Generalized Assignment Problems. *European Journal of Operational Research*, **114**, 165-177.

(20) Parker, R.G. & Rardin, R.L. (1988). *Discrete Optimization*. Academic Press, New York, USA.

(21) Ribeiro, G.M. (2005). Relaxação Lagrangeana com Divisão em Clusters para Problemas de Otimização Modelados em Grafos de Conflitos. Brazilian Space Research Institute – INPE. PhD Project. Available at <http://www.lac.inpe.br/~lorena/glaydston/proposta-glaydston.pdf>.

(22) Strijk, T.; Verweij, B. & Aardal, K. (2000). Algorithms for Maximum Independent Set Applied to Map Labeling. <ftp://ftp.cs.uu.nl/pub/RUU/CStechreps/CS-2000/2000-22.ps.gz>. Accessed November 15, 2003.

(23) Verner, O.V.; Wainwright R.L. & Schoenefeld, D.A. (1997). Placing Text Labels on Maps and Diagrams Using Genetic Algorithms with Masking. *INFORMS Journal on Computing*, **9**, 266-275.

(24) Yamamoto, M.; Camara, G. & Lorena, L.A.N. (2002). Tabu Search Heuristic for Point-Feature Cartographic Label Placement. *GeoInformatica*: *An International Journal on Advances of Computer Science for Geographic Information Systems*. Kluwer Academic Publiser, Netherlands, **6**(1), 77-90.

(25) Yamamoto, M. & Lorena, L.A.N. (2005). A Constructive Genetic Approach to Point-Feature Cartographic Label Placement. **In**: *Metaheuristics: Progress as Real Problem Solvers* [edited by T. Ibaraki, K. Nonobe and M. Yagiura], Kluwer Academic Publishers, 285-300.

(26) Zoraster, S. (1986). Integer Programming Applied to the Map Label Placement Problem. *Cartographica*, **23**(3), 16-27.

(27) Zoraster, S. (1990). The Solution of Large 0-1 Integer Programming Problems Encountered in Automated Cartography. *Operations Research*, **38**(5), 752-759.

(28) Zoraster, S. (1991). Expert Systems and the Map Label Placement Problem. *Cartographica*, **28**(1), 1-9.