Heuristics for Cartographic Label Placement Problems

Glaydston Mattos Ribeiro and Luiz Antonio Nogueira Lorena*

LAC – Computer and Applied Mathematics Laboratory

INPE – Brazilian Space Research Institute

12.227-010, São José dos Campos – SP

Tel: +55 12 3945-6553 FAX: +55 12 3945-6375

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

Abstract

The cartographic label placement problem is an important task in automated cartography and Geographical Information Systems. Positioning the texts requires that overlap among texts should be avoided, that cartographic conventions and preference should be obeyed. This paper examines the point-feature cartographic label placement problem (PFCLP) as an optimization problem. We formulate the PFCLP considering the minimization of existing overlaps and labeling of all points on a map. This objective improves legibility when all points must be placed even if overlaps are inevitable. A new mathematical formulation of binary integer linear programming that allows labeling of all points is presented, followed by some Lagrangean relaxation heuristics. The computational tests considered instances proposed in the literature up to 1000 points, and the relaxations provided good lower and upper bounds.

Keywords: Label placement, Modeling, Lagrangean relaxation, Lagrangean/Surrogate relaxation, Heuristic.

* Corresponding author

1. Introduction

The cartographic label placement problem is an important task in automated cartography and Geographical Information Systems. This problem consists in attach a label, here a text, to each feature point, line or region in the map. Positioning the texts requires that overlap among texts should be avoided, that cartographic conventions and preference should be obeyed. So, the label placement belongs to an area of difficult solution for large instances.

This paper examines the point-feature cartographic label placement problem (PFCLP) with labels restricted to some finite number of positions (discrete model) and being axis-parallel rectangles. If we remove the restriction that a label can be placed at a finite number of positions and allow a continuous movement of the label around of its point (continuous model), we have a model known as slider model (Kreveld et al., 1999; Klau and Mutzel, 2000 and 2003).

Figure 1 - A set of 8 potential label positions and their desirability (Christensen et al., 1995)

The PFCLP can be considered as an optimization problem and has been shown to be NP-Hard (Formann and Wagner, 1991; Marks and Shieber, 1991). The problem consists in placing of point labels in positions in a way that a set of constraints are satisfied, minimizing or maximizing an objective function. Thus, for each point exist a set of possible label positions also known as a set of candidate positions. This set is a list of potential label positions that can indicate their desirability (Christensen et al., 1995). Fig. 1 shows a group of 8 candidate positions for a point, where the numbers indicate the cartographic preference in increasing order.

Placing labels in candidate positions can generate overlaps (conflicts) causing problems in 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 the set of the candidate positions (vertices) and *E* the set of edges representing overlaps or conflicts. Fig. 2B shows the graph of the example in Fig. 2A. This example has two points, each one with 4 candidate positions, the candidate position *L3* has a potential conflict with the position *L6* and *L4* has potential conflicts with *L5* and *L6*.

Figure 2 - Candidate positions, conflicts and the related graph (Yamamoto and Lorena, 2005).

Fig. 3 shows two possible solutions for the same problem with four points. Solution (b) has better visibility than solution (a). Considering these visibility questions, this paper proposes a new integer linear programming model for Minimum Number of Conflicts Problem (MNCP), presenting some Lagrangean and Lagrangean/Surrogate heuristics. This formulation allows labeling all points in a conflict graph. Using the benchmark composed of instances of the literature, a commercial solver could obtain in few seconds the optimal solution for instances up to 500 points, but failed on large instances with 750 and 1000 points, justifying the use of lagrangean heuristics to provide good solutions (upper and lower bonds).

Figure 3 - Possible solutions for a problem with 4 points.

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

2. Literature Review

In the literature the PFCLP have different but connected objectives. The PFCLP can be modeled as a Maximum Independent Set Problem (MISP) (Zoraster, 1990; Strijk et al., 2000) or as a Maximum Number of Conflict Free Labels Problem (MNCFLP) (Christensen, 1993; Yamamoto and Lorena, 2005). Both approaches count the final number of positioned conflict free labels and the main difference is in the final number of labeled points. In MISP, points with inevitable overlaps are not labeled, while all points must be labeled in MNCFLP. For literature on other models, see the map labeling bibliography (Wolf and Strijk, 2005)

Considering the PFCLP as a MISP, several works that propose algorithms and techniques to reduce the number of constraints are found in the literature. 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. Zoraster also used a Lagrangean relaxation and obtained some computational results on small-scale instances. Strijk et al. (2000) proposed other mathematical formulation exploring some cut constraints. These cuts are based on cliques and appeared before in the works of Moon and Chaudhry (1984) and Murray and Church (1996). They applied and proposed several heuristics: Simulated Annealing, Diversified Neighborhood Search, *k-opt* and Tabu Search. The last one showed the better results for their instances. Recently, Klau and Mutzel (2000 and 2003) present exact algorithms where constraint graphs code vertical and horizontal positioning relations, allowing discrete and slider models with integer linear programming.

The Maximum Number of Conflict Free Labels Problem (MNCFLP) was examined in several papers. Hirsch (1982) developed a Dynamic Algorithm of label repulsion, where labels in conflicts are moved trying to remove a conflict. The algorithm defines repelling forces for overlapping labels and computes translation vectors for them. After translation, this process is repeated and hopefully, a labeling with few overlaps appears after a number of iterations. 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 but these algorithms have difficulty of escaping from local maximum. 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 that provides good results when compared with the literature. Schreyer and Raidl (2002) applied Ant Colony System but the results were not interesting when compared to the ones obtained by Yamamoto et al. (2002). Yamamoto and Lorena (2005) developed an exact algorithm for small instances of PFCLP and applied the Constructive Genetic Algorithm (CGA) proposed by Lorena and 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.

The PFCLP, even as a MISP or MNCFLP, can generate a large conflict graph that becomes hard to deal with it. Wagner et al. (2001) presented an approach to reduce the conflict graph provided by a PFCLP. They proposed three rules to reduce the size of the conflict graph without altering the set of optimal solutions. Moreover, they combined these rules with heuristic yielding near-optimal solutions.

3. Mathematical formulation

The mathematical formulation proposed for the MNCP is to minimize the number of conflicts considering that for each point i corresponds a number P_i of candidate positions. 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 of the point i will be used (will receive the label of point i), otherwise, $x_{i,j} = 0$. Besides, for each possible candidate position of point i it is associated a cost (a penalty) represented by $w_{i,j}$.

For each candidate position $x_{i,j}$ exist 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 candidates positions $x_{k,t}$ conflicting with $x_{i,j}$. For all $(k,t) \in S_{i,j}$, where $k \in \{1,...,N\} : k > i$ and $t \in \{1,...,P_k\}$, we have 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) 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 only one candidate position must be selected and so, one of the candidate positions from P_i will be receiving the value 1. This set of constraints can be written as:

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

Placing the labels in a map can generate conflicts (overlaps) with the other labels that are positioned. Thus, we need a new constraint set to represent this case. This set considers each position $x_{i,j}$, its respective conflict positions $x_{k,t}$ and one conflict variable $y_{i,j,k,t}$, expressed as:

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

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

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

Subject to

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

$$x_{i,j} + x_{k,t} - y_{i,j,k,t} \leq 1 \qquad \begin{array}{l} \forall\, i = 1...N \\ \forall\, j = 1...P_i \\ (k,t) \in S_{i,j} \end{array} \tag{6}$$

$$x_{i,j}, x_{k,t} \text{ and } y_{i,j,k,t} \in \{0,1\} \quad \begin{array}{l} \forall\, i = 1...N \\ \forall\, j = 1...P_i \\ (k,t) \in S_{i,j} \end{array} \tag{7}$$

Constraint (7) ensures that all decision variables of the problem are binaries. When the objective function is minimized the conflict variables must be eliminated or minimized (if elimination is not possible).

The formulation (4)-(7) is similar to the one proposed by Zoraster (1990), but in Zorater's formulation points with inevitable overlaps are not labeled and our model ensures that all points are labeled minimizing the number of conflicts.

This formulation was initially tested using the CPLEX 7.5 (ILOG, 2001) on a set of standard problems (Yamamoto and Lorena, 2005; Yamamoto, 2003). The optimal solution could be found in few seconds for the instances up to 500 points. The CPLEX could not obtain the optimal solutions for instances with 750 and 1000 points heuristically solved by Yamamoto et al. (2002) and Yamamoto and Lorena (2005). The CPLEX performed until stop reaching an out of memory condition. Thus, some relaxation heuristics were proposed to provide good lower and upper bounds for MNCP. 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 conflict.

4. Relaxations

We examine in this section three relaxations to the MNCP formulation (4)–(7), a Lagrangean relaxation of constraints (6), a Lagrangean/Surrogate (*LagSur*) relaxation of constraints (6) and a *LagSur* relaxation of constraints (5).

The first relaxation is similar to the one proposed by Zoraster (1990). The model defined in (4)-(7) was solved by a Lagrangean relaxation, with a subgradient optimization and specific heuristic.

The conflict constraints (6) are relaxed in a Lagrangean way. Then, for given multipliers $\lambda \in R_+^C$, where $C = \sum_{i=1}^{N} \sum_{j=1}^{P_i} \left| S_{i,j} \right|$ is the number of the conflict constraints, the Lagrangean relaxation for MNCP (now consider *P* as MNCP) is given by:

$$
\begin{aligned}
v(L^{\lambda} P) = Min \; & \sum_{i=1}^{N} \sum_{j=1}^{P_i} \left(w_{i,j} x_{i,j} + \sum_{(k,t) \in S_{i,j}} y_{i,j,k,t} \right) + \sum_{i=1}^{N} \sum_{j=1}^{P_i} \sum_{(k,t) \in S_{i,j}} \lambda_{i,j,k,t} \left(x_{i,j} + x_{k,t} - y_{i,j,k,t} \right) \\
& - \sum_{i=1}^{N} \sum_{j=1}^{P_i} \sum_{(k,t) \in S_{i,j}} \lambda_{i,j,k,t}
\end{aligned}
\qquad (8)
$$

Subject to *(5) and (7).*

Problem *($L^{\lambda}P$)* is easily solved by inspection over constraint set (5), and for a set of λ values, the best value *$v(L^{\lambda}P)$* is less than or equal to *v(P)* and is obtained solving the Lagrangean dual $\underset{\lambda \geq 0}{Max} \left\{ v\left(L^{\lambda} P\right) \right\}$. To solve it, we use a subgradient algorithm (Parker and Rardin, 1988). At an iteration *l*, the subgradients are defined by $g^{\lambda_l} = x_{i,j}^{\lambda_l} + x_{k,t}^{\lambda_l} - y_{i,j,k,t}^{\lambda_l} - 1$, where $i \in \{1,...,N\}$, $j \in \{1,...,P_i\}$ and $(k,t) \in S_{i,j}$, such that (x^{λ_l} , y^{λ_l}) is an optimal solution to *($L^{\lambda}P$)*. The subgradient method updates the multiplier λ_l as $\lambda_{l+1} = \lambda_l + \theta_l g^{\lambda_l}$, where θ_l is the step size calculated by $\theta_l = \dfrac{\pi\left(ub_l - v(L^{\lambda_l} P)\right)}{\left\| g^{\lambda_l} \right\|^2}$ and ub_l is the value for an improved feasible solution to *P* (to be described in the following) found at iteration *l*. The control of parameter π is the same proposed by Held and Karp (1971), beginning with 2 and halving it whenever ub_l does not decrease for 15 successive iterations. The stopping tests used are: $\pi \leq 0.005$ or $\left(ub_l - v\left(L^{\lambda_l} P\right)\right) < 1$ or $\left\| g^{\lambda_l} \right\|^2 = 0$.

The relaxed solution $Sol_{\mathrm{Re}\,l} = (x^{\lambda}, y^{\lambda})$ of $(L^{\lambda}P)$, composed by variables representing candidate positions ($x^{\lambda}_{i,j}$) and conflicts ($y^{\lambda}_{i,j,k,t}$), is also a feasible solution for P. The conflict constraints (6) are relaxed but constraints (5) are respected. This feasible solution is improved by the following local search heuristic.

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 second relaxation is a Lagrangean/Surrogate (*LagSur*) heuristic (Narciso and Lorena, 1999) over conflict constraints (6). First, as described by Glover (1968), the set (6) is relaxed in the surrogate way followed by a Lagrangean relaxation of the surrogate constraint. For multipliers $\lambda \in R^{C}_{+}$ where C is the number of the conflict constraints defined before, the surrogate relaxation of P is:

$$v(SP_1^{\lambda}) = Min \sum_{i=1}^{N} \sum_{j=1}^{P_i} \left(w_{i,j} x_{i,j} + \sum_{(k,t) \in S_{i,j}} y_{i,j,k,t} \right) \qquad (9)$$

Subject to $$\sum_{i=1}^{N}\sum_{j=1}^{P_i}\sum_{(k,t)\in S_{i,j}}\lambda_{i,j,k,t}\left(x_{i,j}+x_{k,t}-y_{i,j,k,t}\right)\leq\sum_{i=1}^{N}\sum_{j=1}^{P_i}\sum_{(k,t)\in S_{i,j}}\lambda_{i,j,k,t} \quad (10)$$

(5) and (7).

For $t \geq 0$, relaxing constraints (10) of SP_1^{λ} in the Lagrangean way, the *LagSur* relaxation is given by:

$$v(L_t SP_1^{\lambda}) = Min \sum_{i=1}^{N}\sum_{j=1}^{P_i}\left(w_{i,j}x_{i,j}+\sum_{(k,t)\in S_{i,j}}y_{i,j,k,t}\right)+ t.\left(\sum_{i=1}^{N}\sum_{j=1}^{P_i}\sum_{(k,t)\in S_{i,j}}\lambda_{i,j,k,t}\left(x_{i,j}+x_{k,t}-y_{i,j,k,t}\right)-\sum_{i=1}^{N}\sum_{j=1}^{P_i}\sum_{(k,t)\in S_{i,j}}\lambda_{i,j,k,t}\right) \quad (11)$$

Subject to *(5) and (7).*

An interesting characteristic of relaxation $(L_t SP_1^{\lambda})$ is that for $t = 1$ we have the Lagrangean relaxation shown before. For λ fixed, the best value for t can be calculated by solving a *LagSur* dual in variables $t \geq 0$, $v\left(D_t^{\lambda}\right)_1 = \underset{t\geq 0}{Max}\, v\left(L_t SP_1^{\lambda}\right)$.

For λ fixed, the optimal value $v\left(D_t^{\lambda}\right)_1$ provides an improved bound to the Lagrangean relaxation (when t is fixed to 1). Senne and Lorena (2000) described a dichotomous search algorithm that approximates the best value of t. Denoting t^* this best value, if for a number of iterations of the subgradient algorithm the value of the t^* repeats, then t^* is fixed and the dichotomous search is not more executed. For iteration l of the subgradient algorithm, the multipliers are updated as $\lambda_{l+1} = \lambda_l + \theta_l g^{\lambda_l}$, where θ_l is the step size calculated by $\theta_l = \dfrac{\pi\left(ub_l - v(L_{t^*}SP_1^{\lambda_l})\right)}{\left\| g^{\lambda_l}\right\|^2}$, ub_l is the feasible solution found with heuristic IH, and the subgradients are calculated as shown before.

The third relaxation considers a *LagSur* relaxation applied over the constraint set (5). Again, given multipliers $\lambda \in R^N$, where N is the number of points, a surrogate relaxation can be obtained as:

$$v(SP_2^{\lambda}) = Min \sum_{i=1}^{N}\sum_{j=1}^{P_i}\left(w_{i,j}x_{i,j} + \sum_{(k,t)\in S_{i,j}} y_{i,j,k,t} \right) \quad (12)$$

Subject to $$\sum_{i=1}^{N}\sum_{j=1}^{P_i}\lambda_i x_{i,j} = \sum_{i=1}^{N}\lambda_i \quad (13)$$

(6) and (7)

For $t \in R$, constraint (13) of SP_2^{λ} is relaxed in the Lagrangean way giving:

$$v(L_t SP_2^{\lambda}) = Min \sum_{i=1}^{N}\sum_{j=1}^{P_i}\left(w_{i,j}x_{i,j} + \sum_{(k,t)\in S_{i,j}} y_{i,j,k,t} \right) + t.\left(\sum_{i=1}^{N}\sum_{j=1}^{P_i}\lambda_i x_{i,j} - \sum_{i=1}^{N}\lambda_i \right) \quad (14)$$

Subject to (6) and (7)

Now, the subgradient is $g^{\lambda} = \sum_{i=1}^{N}\sum_{j=1}^{P_i} x_{i,j}^{\lambda} - 1$ and the subgradient algorithm updates the multiplier λ_l as before $\lambda_{l+1} = \lambda_l + \theta_l g^{\lambda_l}$, where θ_l is the step size calculated by $\theta_l = \frac{\pi\left(ub_l - v(L_{t*}SP_2^{\lambda_l})\right)}{\left\| g^{\lambda_l} \right\|^2}$, and ub_l is the feasible solution found with the constructive heuristic (CH) described in the following.

The constructive heuristic builds a feasible solution from the relaxed solution $Sol_{Rel} = (x^{\lambda}, y^{\lambda})$ of $(L_t SP_2^{\lambda})$, finding candidate positions (constraints (5)) with smallest number of conflicts related to the feasible solution that is being built. If some point i was not labeled, the heuristic search among the possible candidate positions j of this point, the smallest set $S_{i,j}$ and the respective j is used to label i. The feasible solution obtained is further improved applying heuristic IH.

Constructive Heuristic - CH

Fill the feasible solution array with zeroes;

For $i=1$ to N

Find in relaxed solution all candidate positions different from zero for the point i.

Select for feasible solution in the point 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.

5. Computational Results

The computational tests are performed on instances proposed by Yamamoto and Lorena (2005) that are available at http://www.lac.inpe.br/~lorena/instancias.html. Our implementation used a Pentium IV 2.66GHz processor with 512MB of RAM memory, Windows XP and C++ compiler.

As considered by Zoraster (1990), Yamamoto et al. (2002) and Yamamoto and Lorena (2005), we considered 4 candidate positions and each one has the same desirability ($w_{i,j} = 1$ $\forall$ $i = 1\ldots N$ and $j = 1\ldots P_i$).

Tables 1 to 3 report the average results for $(L^{\lambda}P)$ and $(L_t SP_1^{\lambda})$. Problems with 25 points have 8 instances and all the others 25. The information in columns are:

- *Problem* - Number of points to be labeled;
- *Optimal Solution* - The optimal solution to problem (4) – (7) obtained with CPLEX;
- *Lower Bound* - The best dual limit found;
- *Upper Bound* - The best upper bound (feasible solution) found;
- *Gab_ub* - Percentage deviation from optimal solution to the best upper bound:

$$Gap\_ub = \left(\frac{Upper\ bound - Optimal\ Solution}{Optimal\ Solution} \right) * 100\ ;$$

- G*ap_lb* - Percentage deviation from optimal solution to the best lower bound:

$$Gap\_lb = \left(\frac{Optimal\ Solution - Lower\ bound}{Optimal\ Solution} \right) * 100\ ;$$

- *Iter* - Number of the iterations used by subgradient algorithm;
- *Time* - The total computational time (in seconds).

The search for the best multiplier *t* described by Senne and Lorena (2000) was tending to zero in $(L_t SP_1^{\lambda})$. This fact was expected since all information about conflicts was relaxed. Then, we tested some fixed values for *t:* 0; 0.25; 0.5 and 0.75. Solving $(L_t SP_1^{\lambda})$ for t = 0 is equivalent to a random solution for candidate positions ($x_{i,j}^{\lambda}$) since the conflicts ($y_{i,j,k,t}^{\lambda}$) must be zero in (11).

Problems with 100 and 250 points are simple, and solutions without conflicts are found very quickly and the subgradient algorithm stops on weak lower bounds, consequently the

Gaps are not calculated being substituted by NC (Not Calculated). The same situation also appeared for some instances with 500 points. Thus, the results shown in the Tables 1 to 3 correspond to problems in that the optimization process reaches one of the three stop conditions described in Section 4.

Considering Table 1, the $(L^{\lambda}P)$ lower bound gaps varied from 0.53% to 9.72% and the $(L_t SP_1^{\lambda})$, in the best case, from 0.28% to 9.72%. In Table 2 the $(L^{\lambda}P)$ upper bound gaps varied from 0.00% to 4.14% and the best of $(L_t SP_1^{\lambda})$, from 0.00% to 1.82%. Therefore, in both analysis the $(L_t SP_1^{\lambda})$ provided better results than $(L^{\lambda}P)$. The computational times (see Table 3) were very good for all instances, varying from 0.00 to 19.60 seconds (1000 points).

The results for $(L_t SP_2^{\lambda})$ are reported in Table 4. The CPLEX 7.5 was used to solve the relaxed binary integer linear programs. This relaxation is stronger than $(L^{\lambda}P)$ and $(L_t SP_1^{\lambda})$, however the results were not interesting. Only the lower bound was improved, but the 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 consumed 444 seconds to be completed.

Considering that for large instances we could not ensure optimal solution, the pre-processing suggested by Wagner et al. (2001) is used to reduce the size of conflict graphs. Wagner et al. (2001) developed three rules that can be described as follow. For the first one, if a point p has a candidate position without conflicts, we must use this candidate position as part of the solution and eliminate all other candidates of p. The second, if a point p has a candidate position p_i that is only in conflict with some candidate position q_k, and the point q has a candidate q_j ($j \neq k$) that is only overlapped by p_l ($l \neq i$), then add p_i and q_j to the solution and eliminate all other candidates of p and q. For the last rule, if a point p has only one candidate p_i left, and the candidates overlapping p_i form a clique, then declare p_i to be part of the solution and eliminate all candidates that overlap p_i.

These three rules are applied exhaustively so, after eliminating a candidate p_i, we must check recursively whether the rules can be applied in the neighborhood of p_i. For more details, see Wagner et al. (2001).

Table 5 shows average results obtained with this reduction procedure. The columns indicate the number of vertices ($|V|$), number of edges ($|E|$), number points fixed by procedure (Labeled points), percentage of vertices reduction (Vertices reduction %) and percentage of edges reduction (Edges reduction %). The procedure reduced more edges in instances with 100 and 250 points. For these instances in several cases the reduction procedure provided a solution without overlaps. However, even with reduced graphs, the CPLEX could not found the optimal solutions for larger instances due to out of memory conditions.

Our relaxations are then applied to these new reduced graphs. The results are shown in Tables 6, 7, 8 and 9, where the columns are the same described before. All results were improved, the lower bounds increased and the upper bounds decreased and consequently the gaps reduced. The times are also reduced for the tests, but we found the same situation for the relaxation $(L_t SP_2^{\lambda})$ applied on large instances, i. e., we could not obtain bounds for instances with 500, 750 and 1000 points.

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

6. Conclusion

This paper presented a new mathematical formulation for point-feature cartographic label placement problem that minimizes the number of existing overlaps in a labeling of all points on a map, and is different from the common problem studied in literature that considers the positioning of the maximum number of conflict free labels.

A commercial solver could not solve some large instances proving space to new lagrangean heuristics tested on a set of instances varying from 25 to 1000 points. For many instances the results found are close to the optimal solutions, providing good lower and upper bounds.

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

Acknowledgements - The authors acknowledge the useful comments and suggestions of two referees. This research was partially spported by CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico.

References

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.

Christensen, J., Marks, J., Shieber, S., 1993. Placing text labels on maps and diagrams. London. Academic Press, Harvard University, 10 pp.

Formann, M., Wagner, F., 1991. A packing problem with applications to lettering of maps. In: Proceedings of the 7th Annual Symposium on Computational Geometry, New Hampshire, USA, pp. 281-288.

Glover, F., 1968. Surrogate constraints. Operations Research 16, 741-749.

Held, M., Karp, R. M., 1971. The traveling salesman problem and minimum spanning trees: Part II. Mathematical Programming 1, 6-25.

Hirsch, S. A., 1982. An algorithm for automatic name placement around point data. American Cartographer 9 (1), 5-17.

ILOG, 2001. Ilog Cplex 7.5 Reference Manual. ©Copyright 2001 by Ilog, 358 pp.

Klau, G. W., Mutzel, P., 2000. Optimal labeling of point features in the slider model. In Du, D.-Z., Eades, P., Estivill-Castro, V., Lin, X., Sharma, A. (Eds), *Proc. 6th Annual International Computing and Combinatorics Conf. (COCOON'00)*, volume 1858 of *Lecture Notes in Computer Science*, 340-350. Springer-Verlag.

Klau, G. W., Mutzel, P., 2003. Optimal labeling of point features in rectangular labeling models. Mathematical Programming Ser. B 94, 435-458.

Kreveld, M. V., Strijk, T., Wolff, A., 1999. Point labeling with sliding labels. Computational Geometry 13, 21-47.

Lorena, L. A. N., Furtado, J. C., 2001. Constructive genetic algorithm for clustering problems. Evolutionary Computation 9 (3), 309-327.

Marks, J., Shieber, S., 1991. The computational complexity of cartographic label placement. Advanced Research in Computing Technology, Harvard University, Massachusets, 28 pp.

Moon, I. D., Chaudhry, S., 1984. An analysis of network location problems with distance constraints. Management Science 30, 290-307.

Murray, A. T., Church, R. L., 1996. Solving the anti-covering location problem using lagrangian relaxation. Computers & Operations Research 24(2), 127-140.

Narciso, M. G., Lorena, L. A. N., 1999. Lagrangean/Surrogate relaxation for generalized assignment problems. European Journal of Operational Research 114, 165-177.

Parker, R. G., Rardin, R. L., 1988. Discrete optimization, 1st edn. Academic Press, New York, 472pp.

Schreyer, M., Raidl, G. R., 2002. Letting ants labeling point features. In: Proceedings of the 2002 IEEE Congress on Evolutionary Computation at the IEEE World Congress on Computational Intelligence, pp. 1564-1569.

Senne, E. L. F., Lorena, L., A., N, 2000. Lagrangean/Surrogate heuristics for p-median problems. In *Computing Tools for Modeling, Optimization and Simulation*. In: Laguna, M. and Gonzalez-Velarde, J. L. (eds). Kluwer Academic Publishers, New York, 115-130.

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.

Wagner, F., Wolff, A., Kapoor, V., Strijk, T., 2001. Three rules suffice for good label placement. Algorithmica 30, 334-349.

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.

Yamamoto, M., 2003. Novos algoritmos para rotulação cartográfica de pontos (New algorithms for point-feature cartographic label placement). Ph.D. Dissertation, INPE - Brazilian Space Research Institute, São José dos Campos, São Paulo,114 pp.

Yamamoto, M., Lorena, L. A. N., 2005. A constructive genetic approach to point-feature cartographic label placement. In: Ibaraki, T., Nonobe, K. and Yagiura, M. (eds). *Metaheuristics: Progress as Real Problem Solvers*. Kluwer Academic Publishers. (To appear).

Zoraster, S., 1986. Integer programming applied to the map label placement problem. Cartographica 23 (3), 16-27.

Zoraster, S., 1990. The solution of large 0-1 integer programming problems encountered in automated cartography. Operations Research 38 (5), 752-759.

Zoraster, S., 1991. Expert systems and the map label placement problem. Cartographica 28 (1), 1-9.

Internet References

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. 42 pp.

Wolf, A., Strijk, T., (2005). The map labeling bibliography. URL on June 6th: http://i11www.ilkd.uni-karlsruhe.de/~awolff/map-labeling/bibliography/

Figure Captions

Figure:

1 – A set of 8 potential label positions and their desirability (Christensen et al., 1995).

2 – Candidate positions, conflicts and the related graph (Yamamoto and Lorena, 2005).

3 – Possible solutions for a problem with 4 points.

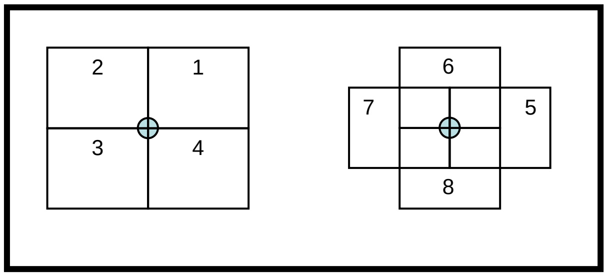

Figure 1

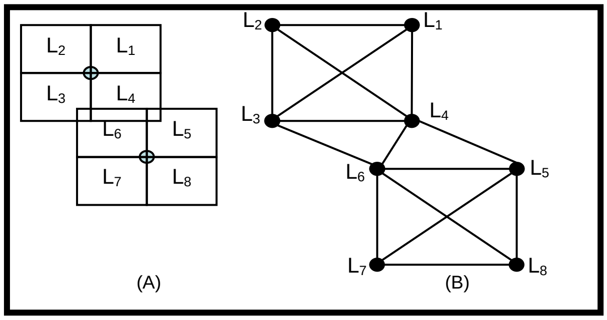

Figure 2

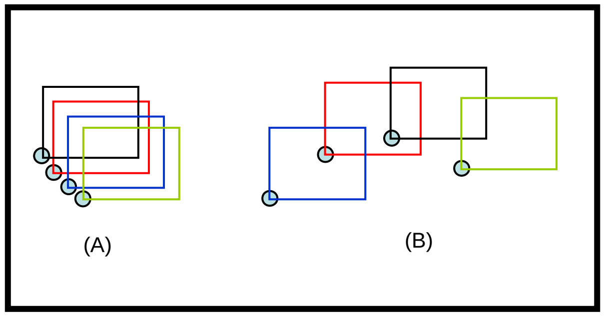

Figure 3

Table Captions

Table:

1 – Lower Bound Average Results for Lagrangean and *LagSur* Relaxation over Constraint Set (6).

2 – Upper Bound Average Results for Lagrangean and *LagSur* Relaxation over Constraint Set (6).

3 – Iterations and Computational Times for Lagrangean and *LagSur* Relaxation over Constraint Set (6).

4 – Average results obtained for $(L_t SP_2^{\lambda})$ – Bounds, *Gaps*, Iterations and Computational Times.

5 – Average results on reduced conflict graphs obtained after rules proposed by Wagner et al. (2001).

6 – Lower bound average results on reduced graphs.

7 – Upper bound average results on reduced graphs.

8 – Iterations and computational times on reduced graphs.

9 – Average results obtained for $(L_t SP_2^{\lambda})$ on reduced graphs.

Problem	Optimal Solution	Lower bound				Gap_lb(%)			
		$(L^{\lambda}P)$	$(L_t SP_1^{\lambda})$			$(L^{\lambda}P)$	$(L_t SP_1^{\lambda})$		
			t=0.25	t=0.50	t=0.75		t=0.25	t=0.50	t=0.75
25.00	27.75	25.00	24.96	25.00	25.00	9.72	9.87	9.72	9.72
100.00	100.00	NC	NC	NC	NC	NC	NC	NC	NC
250.00	250.00	NC	NC	NC	NC	NC	NC	NC	NC
500.00	500.84	498.20	493.46*	496.91*	499.45	0.53	1.48*	0.80*	0.28
750.00	-	749.98	743.42	748.74	749.98	-	-	-	-
1000.00	-	999.93	997.04	999.93	999.95	-	-	-	-

*Average between results that presented solution with conflicts.

Table 1

Problem	Optimal Solution	Upper bound				Gap_ub(%)			
		$(L^{\lambda}P)$	$(L_t SP_1^{\lambda})$			$(L^{\lambda}P)$	$(L_t SP_1^{\lambda})$		
			t=0.25	t=0.50	t=0.75		t=0.25	t=0.50	t=0.75
25.00	27.75	28.88	28.38	28.25	28.63	4.14	2.33	1.82	3.21
100.00	100.00	100.00	100.00	100.00	100.00	0.00	0.00	0.00	0.00
250.00	250.00	250.00	250.00	250.04	250.00	0.00	0.00	0.02	0.00
500.00	500.84	503.92	502.88*	503.16*	503.80	0.61	0.41*	0.46*	0.59
750.00	-	774.44	771.80	772.68	774.44	-	-	-	-
1000.00	-	1086.44	1075.64	1079.24	1086.40	-	-	-	-

* Average between results that presented solution with conflicts.

Table 2

Problem	Iterations				Time (sec.)				
	$(L^{\lambda}P)$	$(L_t SP_1^{\lambda})$			Optimal Solution	$(L^{\lambda}P)$	$(L_t SP_1^{\lambda})$		
		t=0.25	t=0.50	t=0.75			t=0.25	t=0.50	t=0.75
25.00	146.63	148.88	148.38	146.63	1.60	0.25	0.13	0.13	0.13
100.00	1.16	1.16	1.16	1.16	0.02	0.00	0.00	0.00	0.04
250.00	2.48	2.36	7.72	2.68	0.06	0.00	0.00	0.04	0.00
500.00	146.00	140.24	134.48	146.20	3.12	2.76	2.64	2.64	2.80
750.00	146.00	146.00	146.00	146.00	-	8.48	8.44	8.40	8.56
1000.00	146.00	146.00	146.00	146.00	-	19.32	19.60	19.52	19.08

Table 3

Instance	Optimal solution	Lower bound	Upper bound	Gap_ub (%)	Gap_lb (%)	Iter	Time (s)	
							Optimal solution	$(L_t SP_2^{\lambda})$
25	27.75	25.13	28.38	2.29	9.27	151.63	1.60	104.63
100	100.00	NC	100.00	0.00	NC	1.00	0.02	0.16
250	250.00	NC	250.00	0.00	NC	1.48	0.06	0.92

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

Table 4

Instance	Original graph		Reduced graph		
	$\|V\|$	$\|E\|$	Labeled points	Vertices reduction (%)	Edges reduction (%)
25	100	332.38	4.63	18.50	11.65
100	400	102.28	98.52	98.52	90.66
250	1000	614.00	229.60	91.84	74.95
500	2000	2409.44	344.36	68.87	46.43
750	3000	5431.68	347.84	46.38	29.08
1000	4000	9700.92	276.36	27.64	16.24

Table 5

Problem	Optimal Solution	Lower bound				Gap_lb(%)			
		$(L^{\lambda}P)$	$(L_t SP_1^{\lambda})$			$(L^{\lambda}P)$	$(L_t SP_1^{\lambda})$		
			t=0.25	t=0.50	t=0.75		t=0.25	t=0.50	t=0.75
25.00	27.75	25.00	24.94	25.00	25.00	9,72	9,95	9,72	9,72
100.00	100.00	NC	NC	NC	NC	NC	NC	NC	NC
250.00	250.00	NC	NC	NC	NC	NC	NC	NC	NC
500.00	500.84	499.65*	497.77*	499.29*	499.92*	0.24*	0.63*	0.33*	0.19*
750.00	-	749.98	747.34	749.76	749.98	-	-	-	-
1000.00	-	999.93	998.67	999.96	999.94	-	-	-	-

*Average between results that presented solution with conflicts.

Table 6

Problem	Optimal Solution	Upper bound				Gap_ub(%)			
		$(L^{\lambda}P)$	$(L_t SP_1^{\lambda})$			$(L^{\lambda}P)$	$(L_t SP_1^{\lambda})$		
			t=0.25	t=0.50	t=0.75		t=0.25	t=0.50	t=0.75
25.00	27.75	28.25	28.25	28.50	28.25	1.81	1.81	2.70	1.81
100.00	100.00	100.00	100.00	100.00	100.00	0.00	0.00	0.00	0.00
250.00	250.00	250.00	250.00	250.04	250.00	0.00	0.00	0.02	0.00
500.00	500.84	502.80	502.88	503.28	503.04	0.39	0.41	0.49	0.44
750.00	-	772.36	770.20	772.16	772.36	-	-	-	-
1000.00	-	1081.12	1074.76	1077.24	1081.16	-	-	-	-

* Average between results that presented solution with conflicts.

Table 7

Problem	Iterations				Time (sec.)				
	$(L^{\lambda}P)$	$(L_t SP_1^{\lambda})$			Optimal Solution	$(L^{\lambda}P)$	$(L_t SP_1^{\lambda})$		
		t=0.25	t=0.50	t=0.75			t=0.25	t=0.50	t=0.75
25.00	147.38	149.50	147.00	148.25	0.44	0.13	0.13	0.25	0.25
100.00	1.04	1.04	1.04	1.04	0,20	0.08	0.04	0.04	0.01
250.00	8.36	1.64	10.16	2.68	0,16	0,04	0	0,04	0,00
500.00	140.48	134.40	128.76	140.24	0.61	1.52	1.44	1.36	1.44
750.00	146.00	146.00	146.00	146.00	-	6.12	6.12	6.20	6.04
1000.00	146.00	146.00	146.00	146.00	-	16.00	16.76	16.12	16.08

Table 8

Instance	Optimal solution	Lower bound	Upper bound	Gap_ub (%)	Gap_lb (%)	Iter	Time (s)	
							Optimal solution	$(L_t SP_2^{\lambda})$
25	27.75	25.13	28.25	1.81	9.26	147.63	0.44	95,00
100	100.00	NC	100,00	0,00	NC	1,00	0.20	0,08
250	250.00	NC	250,04	0,02	NC	5,64	0.16	0,40

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

Table 9