



## HEURÍSTICA GULOSA PARA O PROBLEMA DA ROTULAÇÃO CARTOGRÁFICA DE PONTOS

Gildásio Lecchi Cravo<sup>1</sup>, Glaydston Mattos Ribeiro<sup>1,2</sup> e  
Luiz Antonio Nogueira Lorena<sup>2</sup>

<sup>1</sup> Departamento de Ciência da Computação e Informática  
Faculdade de Aracruz – Uniaracruz  
{lecchi, glaydston}@fsjb.edu.br

<sup>2</sup> Laboratório Associado de Matemática e Computação Aplicada  
Instituto Nacional de Pesquisas Espaciais – INPE  
{glaydston, lorena}@lac.inpe.br

### RESUMO

O problema da rotulação cartográfica de pontos consiste em rotular os pontos de um mapa evitando as sobreposições dos rótulos. Este trabalho apresenta uma heurística gulosa para esse problema, baseada no grafo de conflitos produzido. Resultados computacionais mostram a eficiência dessa heurística, superando diversos algoritmos e metaheurísticas em um tempo computacional baixo.

**Palavras-chave:** Rotulação cartográfica de pontos. Algoritmo guloso. Grafo de conflitos.

### ABSTRACT

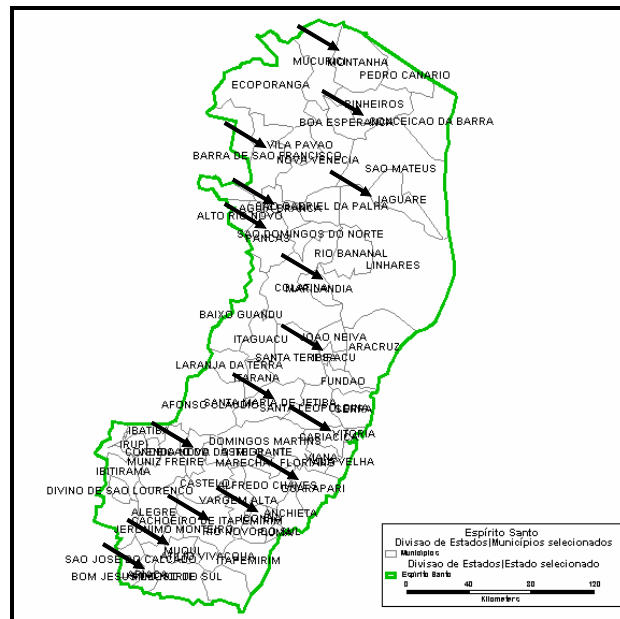
This paper deals with the point-feature label placement problem. The objective of this problem consists in placing the labels in the map avoiding overlaps. The paper presents a greedy heuristic that is based on conflict graphs. Computational results show that this heuristic solves efficiently instances proposed in the literature, producing better results than several heuristics and metaheuristics in excellent runtime performance.

**Keywords:** Point-feature label placement. Greedy algorithm. Conflict graph.

## INTRODUÇÃO

O Problema da Rotulação Cartográfica de Pontos (PRCP) consiste em rotular os pontos de um mapa evitando as sobreposições dos rótulos (RIBEIRO, 2005), ou seja, dado um conjunto de pontos a serem rotulados, o PRCP procura posicionar os rótulos dos pontos fornecendo a melhor visualização possível, com um baixo tempo computacional.

Devido ao crescimento das aplicações Desktop e voltadas para WEB, que permitem mostrar e tomar decisões baseadas em mapas georreferenciados, surge a necessidade de desenvolver algoritmos eficientes que permitam posicionar os textos em um mapa, evitando conflitos e, conseqüentemente, mapas “poluídos” visualmente, conforme pode ser visto na Figura 1.



**Figura 1. Rotulação Cartográfica de Pontos**

**Fonte: Ribeiro (2005)**

Sendo assim, este trabalho apresenta uma heurística gulosa, baseada em grafos de conflitos, que produz bons resultados quando aplicada em algumas instâncias presentes na literatura. Essa heurística apresentou um baixo tempo computacional, mesmo para as maiores instâncias, e ainda rotulações de boa qualidade, superando vários algoritmos, como a Busca Tabu, de Yamamoto et al. (2002), e os resultados médios do Algoritmo Genético Construtivo (AGC), de Yamamoto e Lorena (2005).

Neste trabalho, é feita uma revisão bibliográfica do PRCP assim como uma definição de grafo de conflitos. Em seguida, é apresentado o algoritmo guloso desenvolvido e, por fim, são indicados os resultados computacionais encontrados, as conclusões e as referências.

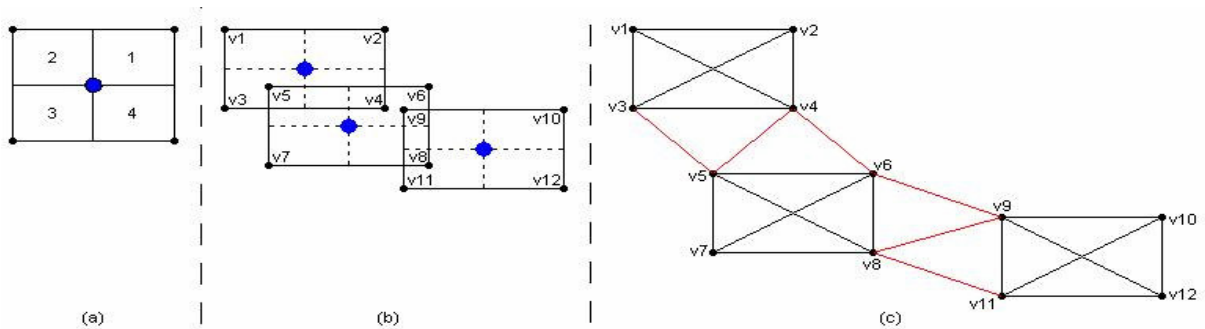
## REVISÃO BIBLIOGRÁFICA

Na literatura, existem três diferentes problemas de rotulação: rotulação de pontos (ex.: cidades, estações de trem, etc.), linhas (rodovias, rios, etc.) e área (Estados, municípios, etc.). Como mencionado, este trabalho está voltado para a rotulação de pontos.

Entretanto, ao posicionar o rótulo de um ponto, deve-se observar a padronização cartográfica que limita as possíveis posições para o rótulo. Sendo assim, uma vez definida a padronização cartográfica, o PRCP pode ser visualizado como um problema de otimização combinatória. A padronização cartográfica mais conhecida na literatura é a de Christensen et al. (1995) e pode ser vista na Figura 2(a). Essa padronização indica locais preferenciais, ou seja, na Figura 2(a), quanto menor for o número associado à posição cartográfica, mais preferencial é a posição.

Outro detalhe importante no PRCP é que apenas uma posição candidata deve receber o rótulo do ponto, com isso cada posição candidata pode representar um vértice em um grafo e as arestas, as possíveis sobreposições dos rótulos. Para um melhor entendimento, considere a Figura 2(b). Nessa figura, existem três pontos, cada um com quatro posições candidatas. Note que existem determinadas áreas sobrepostas. Transformando o problema em um grafo, tem-se a Figura 2(c). Nesse grafo, as arestas  $\{(V_3V_5), (V_4V_5), (V_4V_6), (V_6V_9), (V_8V_9), (V_8V_{11})\}$  indicam possíveis sobreposições (conflitos) entre posições candidatas de pontos diferentes.

O grafo mostrado na Figura 2(c) também é conhecido na literatura por grafo de conflitos (YAMAMOTO et al., 2002). Um grafo de conflitos representa relações lógicas entre variáveis binárias (ATAMTÜRK et al., 2000).



**Figura 2 (a) - Padronização cartográfica segundo Christensen et al. (1995), (b) - exemplo de um problema e (c) - grafo de conflitos**

Considerando essa representação em grafo de conflitos e que o PRCP é classificado como *NP-Hard* (MARKS; SCHEIBER, 1991), vários algoritmos foram propostos na literatura. Entretanto existem três abordagens para esse problema. A primeira consiste em obter o máximo número de rótulos posicionados sem conflitos, nesse caso pontos podem ficar sem rótulos. Essa abordagem também é conhecida como o problema do máximo conjunto independente de vértices. A segunda considera o máximo número de rótulos sem conflitos em uma rotulação de todos os pontos. Por último, a terceira consiste em minimizar o número de conflitos em uma rotulação de todos os pontos. Dado que os melhores resultados da literatura consideram essa terceira abordagem, este trabalho também está direcionado para ela.

Na literatura, existem algoritmos que utilizam relaxação lagrangiana (ZORASTER, 1991) e relaxação de programação linear com *branch-and-bound* (STRIJK et al., 2000). Strijk et al. (2000) ainda desenvolveram e utilizaram diversas heurísticas. Entre elas: *Simulated Annealing*, Busca em Vizinhança Diversificada, *k-Opt* e Busca Tabu.

Christensen et al. (1995) propuseram um método de busca exaustiva. Os autores ainda sugeriram um Algoritmo Guloso e um algoritmo denominado *Discrete Gradient Descent*.

Verner et al. (1997) aplicaram um Algoritmo Genético. Mais tarde, Yamamoto et al. (2002) usaram uma Busca Tabu e Yamamoto e Lorena (2005) usaram um Algoritmo Genético Construtivo.

Recentemente Ribeiro e Lorena (2005) apresentaram uma formulação matemática para o problema da minimização do número de conflitos (PMNC). A formulação está descrita abaixo:

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

Sujeito a:

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

$$\left| C_{i,j} \right| x_{i,j} + \sum_{(k,t) \in S_{i,j}} x_{k,t} - \sum_{c \in C_{i,j}} y_{i,j,c} \leq \left| C_{i,j} \right| \quad \forall i = 1 \dots N \quad (3)$$

$$\forall j = 1 \dots P_i$$

$$x_{i,j} \text{ e } y_{i,j,c} \in \{0,1\} \quad \forall i = 1 \dots N \quad (4)$$

$$\forall j = 1 \dots P_i$$

$$c \in C_{i,j}$$

Sendo:

- $x_{i,j}$  variável binária tal que  $i \in N$  e  $j \in P_i$ , sendo  $P_i$  o número de posições candidatas para o ponto  $i$  e  $N$  o número de pontos a serem rotulados;
- $w_{i,j}$  um custo associado a cada posição candidata  $x_{i,j}$ , o que permite priorizar algumas posições candidatas (padronização candidata);
- $S_{i,j}$  um conjunto de pares de índices  $(k,t): k > i$  de posições candidatas  $x_{k,t}$  conflitantes com  $x_{i,j}$ ;
- $C_{i,j}$  um conjunto com todos os pontos que possuem posições candidatas em conflito com a posição candidata  $x_{i,j}$ , e  $y_{i,j,c}$  uma variável de conflito entre a posição candidata  $x_{i,j}$  e o ponto  $c \in C_{i,j}$ :  $c > i$ .

Ribeiro e Lorena (2005) apresentaram uma relaxação lagrangiana para a modelagem acima, proporcionando os melhores resultados da literatura até então.

Por outro lado, o grafo de conflitos para o PRCP pode ser grande, dificultando o processo de solução. Wagner et al. (2001) apresentaram um método para reduzir o grafo de conflitos composto por três regras. A primeira regra diz que, se um ponto  $p$  tem uma posição candidata livre de conflitos, deve-se usar essa posição candidata na solução do problema e todas as outras posições candidatas do ponto  $p$  devem

ser eliminadas. A segunda diz que, se um ponto  $p$  tem uma posição candidata  $p_i$  que está em conflito somente com uma posição candidata  $q_k$ , e o ponto  $q$  tem uma posição candidata  $q_j$  ( $j \neq k$ ) que está sobreposta somente por uma posição candidata  $p_l$  ( $l \neq i$ ), então, deve-se adicionar  $p_i$  e  $q_j$  na solução do problema e eliminar as demais posições candidatas de  $p$  e  $q$ . A última regra diz que, se um ponto  $p$  tem somente uma posição candidata  $p_i$  restante, e todas as posições candidatas que sobrepõem  $p_i$  formam uma clique, então, deve-se adicionar  $p_i$  na solução do problema e eliminar todas as posições candidatas que sobrepõem  $p_i$ .

Essas regras são aplicadas exaustivamente, ou seja, após a eliminação de uma posição candidata  $p_i$ , deve-se verificar recursivamente se as regras podem ser aplicadas na vizinhança de  $p_i$ . Essa redução apresenta bons resultados. Para maiores detalhes do algoritmo, ver Wagner et al. (2001).

## HEURÍSTICA GULOSA PARA O PRCP

A heurística gulosa proposta é baseada na heurística RSF (*Recursive-Smallest-First*) de Yamamoto (2003) e utiliza a técnica de redução proposta por Wagner et al. (2001). Dado o grafo de conflitos, a heurística RSF inicia escolhendo o vértice de grau mínimo para entrar na solução do problema, em seguida, esse vértice e seus adjacentes são colocados como inativos. Conseqüentemente, os vértices ativos que apresentavam conflitos com os vértices removidos devem ter os graus recalculados. Em seguida, um novo vértice de grau mínimo é selecionado e o ciclo é, então, repetido. O algoritmo termina quando não houver mais vértices ativos. Todos os vértices escolhidos formam um conjunto independente de vértices.

A RSF é uma heurística interessante e pode ser utilizada no PRCP, quando este puder ser modelado conforme a primeira abordagem descrita na seção anterior que tenta rotular o maior número possível de pontos sem conflitos. Sendo assim, para considerar a estratégia da minimização do número de conflitos, esse algoritmo precisa ser modificado para evitar, mas permitir, se necessário, que vértices em conflitos sejam selecionados. Isso porque a abordagem de minimização do número de conflitos rotula todos os pontos mesmo que conflitos sejam inevitáveis.

Portanto, o algoritmo guloso proposto considera como critério de ordenação não só o grau de cada vértice, mas também a existência de possíveis conflitos com a solução que está sendo construída.

Uma vez definido o vértice  $x_{i,j}$  de menor grau, devem-se tornar inativos somente os demais vértices do ponto  $i$ , inclusive o  $x_{i,j}$ , diferentemente da heurística RSF. Além disso, ao calcular o grau dos vértices ativos, no caso dos vértices não removidos adjacentes ao vértice  $x_{i,j}$  selecionado, acrescenta-se uma constante dada por  $N \cdot P_i$  ao valor obtido para o grau. Isso evita que esses vértices sejam escolhidos no próximo passo, ou seja, na seleção do próximo vértice de grau mínimo. No entanto, à medida que a heurística vai sendo executada, esses vértices poderão ser escolhidos no final do processo.

O algoritmo guloso utiliza duas listas dinâmicas  $L1$  e  $L2$ , que representam, respectivamente, a lista de vértices ativos e a lista de vértices inativos. Na definição do vértice de menor grau, uma vez definido o grau acrescido da constante, faz-se uma ordenação crescente baseada nesses valores de grau e escolhe-se o primeiro vértice. Em seguida, esse vértice e os demais que deverão ser removidos do grafo de conflitos são inseridos no final da lista de vértices inativos. Assim, dado que os vértices são inseridos em uma ordem diferente da lista inicial de vértices ativos, ao término da heurística gulosa, a lista de vértices inativos passa a possuir uma ordenação diferente.

Experimentos computacionais mostraram que uma reaplicação da heurística gulosa, sobre essa lista de vértices inativos transformada em vértices ativos, permite melhorar os resultados. Assim, a heurística gulosa aqui proposta pode ser descrita conforme o seguinte pseudocódigo dentro de um laço de repetição, de modo a obter resultados melhores. No algoritmo abaixo,  $N\_Iterações$  representa o número máximo de repetições do algoritmo guloso.

Recentemente Yamamoto et al. (2005) apresentaram uma heurística denominada FALP (*fast algorithm for label placement*) que utiliza a heurística RSF seguida de uma heurística de busca local. Assim, após a aplicação da RSF, a heurística de busca local procura rotular os pontos ainda não rotulados, maximizando o número de rótulos livres de conflitos (segunda abordagem). A heurística gulosa proposta neste trabalho difere da FALP de Yamamoto et al. (2005), por utilizar uma técnica de

redução do grafo de conflitos e pela sua abordagem que é voltada para a minimização do número de conflitos.

### Algoritmo

```

i = 1
S* = {}
Sol* = N*(Maior Pi ∀ i = 1...N)
L1 = {x1,1, x1,2, ..., xN,PN}
Aplicar técnica de redução proposta por Wagner et al. (2001)
Enquanto (i ≤ N_Iteracoes) Faça
    L2 = {}
    S = {}
    Enquanto (L1 ≠ {}) Faça
        Ordene crescentemente L1
        Selecione o primeiro vértice v de L1
        S = S + {v}
        Executar processo de remoção em L1 e atualizar L2;
    Fim Enquanto
    Solução_de_S = Função Objetivo (S)
    Se (Solução_de_S < Sol*) Então
        Sol* = Solução_de_S;
        S* = S;
    Fim Se
    L1 = L2;
    i = i + 1;
Fim Enquanto
Retornar S* e Sol*

```

## RESULTADOS COMPUTACIONAIS

Os testes computacionais foram realizados sobre as instâncias propostas por Yamamoto et al. (2002) disponíveis em <http://www.lac.inpe.br/~lorena/instancias.html>. A heurística gulosa foi implementada em C++, e os testes computacionais foram executados em um PC com processador Athlon™ XP 2800+ (~2,2 GHz) com 256MB de memória RAM. Como realizado em Christensen et al. (1995), Verner et al. (1997), Yamamoto e Lorena (2005) e Ribeiro e Lorena (2005), para todos os problemas não foram consideradas preferências cartográficas, ou seja, custo ou penalidade igual a 1 para todas as posições candidatas, sendo o número dessas posições igual a 4 para todos os pontos.

A Tabela 1 apresenta os resultados computacionais médios para cada problema (25, 100, 250, 500, 750 e 1000 pontos) obtidos pela Heurística Gulosa (HG) proposta e



pela relaxação lagrangiana com clusters (LagClus) proposta por Ribeiro e Lorena (2005). O problema com 25 pontos possui 8 instâncias, enquanto os demais apresentam 25 instâncias cada um. O número máximo de iterações utilizado no algoritmo guloso foi:  $N_{\text{Iterações}} = 35$ .

**Tabela 1. Comparação dos resultados obtidos com os de Ribeiro e Lorena (2005)**

Problema	LagClus* (RIBEIRO; LORENA, 2005)		Heurística Gulosa		Diferença
	Solução	Tempo (s)	Solução	Tempo (s)	
25	28,13	3,50	28,15	0,003	0,02
100	100,00	0,12	100,00	0,001	0,00
250	250,00	0,12	250,00	0,013	0,00
500	500,84	0,40	500,96	0,196	0,12
750	758,96	53,84	761,60	1,738	2,64
1000	1047,32	3445,40	1049,96	10,441	2,64

\* Máquina usada por Ribeiro e Lorena (2005): Pentium IV 2,66GHz, com 512MB de memória RAM

Note que, no pior caso, o algoritmo proposto apresenta um erro de no máximo 2,64 conflitos em média a mais. Outro detalhe importante é o tempo computacional que foi muito inferior ao tempo utilizado pela relaxação de Ribeiro e Lorena (2005).

A Tabela 2 vem sendo utilizada por diversos autores para comparar os resultados dos diferentes algoritmos. Essa tabela informa o percentual médio de rótulos livres de conflitos. Observe que mais uma vez os resultados foram bem interessantes. O algoritmo proposto supera diversos algoritmos importantes na literatura, como os resultados médios do Algoritmo Genético Construtivo (AGC) de Yamamoto e Lorena (2005) e a Busca Tabu de Yamamoto et al. (2002). Comparando os resultados obtidos com os melhores produzidos pelo AGC, somente para as instâncias com 750, os resultados não foram superiores, fornecendo uma média igual a 96,99% contra 97,10% do AGC.

Por outro lado, os resultados encontrados pelo algoritmo guloso estão muito próximos dos resultados obtidos por Ribeiro e Lorena (2005), conforme pode ser visto na tabela.

Tabela 2. Comparação dos resultados obtidos com a literatura

Algoritmo	Percentual de Rótulos Livres				
	Problemas				
	100	250	500	750	1000
LagClus (RIBEIRO; LORENA, 2005)	100,00	100,00	99,67	97,65	91,42
Algoritmo Guloso Proposto	100,00	100,00	99,62	96,99	90,74
Algoritmo Genético Construtivo <sub>Melhor</sub> (YAMAMOTO; LORENA, 2005)	100,00	100,00	99,60	97,10	90,70
FALP (YAMAMOTO et al., 2005)	100,00	100,00	99,50	96,70	90,12
Algoritmo Genético Construtivo <sub>Média</sub> (YAMAMOTO; LORENA, 2005)	100,00	100,00	99,60	96,80	90,40
Busca Tabu (YAMAMOTO et al. 2002)	100,00	100,00	99,30	96,80	90,00
Algoritmo Genético com Máscara (VERNER et al., 1997)	100,00	99,98	98,79	95,99	88,96
Algoritmo Genético (VERNER et al., 1997)	100,00	98,40	92,59	82,38	65,70
Simulated Annealing (CHRISTENSEN et al., 1995)	100,00	99,90	98,30	92,30	82,09
Zoraster (1991)	100,00	99,79	96,21	79,78	53,06
Hirsh (1982)	100,00	99,58	95,70	82,04	60,24
3-opt Gradient Descent (CHRISTENSEN et al., 1995)	100,00	99,76	97,34	89,44	77,83
2-opt Gradient Descent (CHRISTENSEN et al., 1995)	100,00	99,36	95,62	85,60	73,37
Gradient Descent (CHRISTENSEN et al., 1995)	98,64	95,47	86,46	72,40	58,29
Algoritmo Guloso (CHRISTENSEN et al., 1995)	95,12	88,82	75,15	58,57	43,41

## CONCLUSÕES E RECOMENDAÇÕES

Este trabalho apresentou uma heurística gulosa para o problema da rotulação cartográfica de pontos. Os resultados foram bem satisfatórios, superando heurísticas e metaheurísticas conhecidas. Os tempos computacionais também foram baixos, fortalecendo a sua importância.

Dado que o algoritmo aqui proposto possui uma forte ligação com o GRASP (FEO et al., 1994), recomenda-se estudar uma heurística de busca local e implementar um GRASP específico para o PRCP. Devido aos resultados obtidos neste trabalho, espera-se que as soluções sejam ainda melhores.

## REFERÊNCIAS

- 1 ATAMTÜRK, A.; NEMHAUSER, G. L.; SAVELSBERGH, M. W. P. Conflict graphs in solving integer programming problems. **European Journal of Operational Research**, v. 121, p. 40-55, 2000.
- 2 CHRISTENSEN J.; MARKS J.; SHIEBER S. An empirical study of algorithms for point-feature label placement. **ACM Transactions on Graphics**, v. 14, n. 3, p. 203-232, 1995.

- 3 FEO, T. A.; RESENDE, M. G. C.; SMITH, A. A greedy randomized adaptive search procedure for maximum independent set. **Operations Research**, v. 42, p. 860-878, 1994.
- 4 HIRSCH, S. A. An algorithm for automatic name placement around point data. **American Cartographer**, v. 9, n. 1, p. 5-17, 1982.
- 5 MARKS, J.; SCHEIBER, S. The computational complexity of cartographic label placement. **Center for Researching Computing Technology**, Harvard University, 1991.
- 6 RIBEIRO, G. M. Relaxação lagrangiana com divisão em clusters para alguns problemas de otimização modelados em grafos de conflitos. Proposta de Tese de Doutorado em Computação Aplicada, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2005.
- 7 RIBEIRO, G. M.; LORENA, L. A. N. Lagrangean relaxation with clusters for point-feature cartographic label placement problems. **Computers and Operations Research**, 2005, To Appear.
- 8 STRIJK T.; VERWEIJ B.; AARDAL K. Algorithms for maximum independent set applied to map labeling. Disponível em: <ftp://ftp.cs.uu.nl/pub/RUU/CStechreps/CS-2000/2000-22.ps.gz>. Acesso em: 10 nov. 2005.
- 9 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**, n. 9, p. 266-275, 1997.
- 10 WAGNER, F.; WOLFF, A.; KAPOOR, V.; STRIJK, T. Three rules suffice for good label placement. **Algorithmica**, n. 30, p. 334-349, 2001.
- 11 YAMAMOTO M.; CAMARA G.; LORENA L.A.N. Tabu search heuristic for point-feature cartographic label placement. **Geoinformatica and International Journal on Advances of Computer Science for Geographic Information Systems**, Kluwer Academic Publisher, Netherlands, v. 6, n. 1, p. 77-90, 2002.
- 12 YAMAMOTO, M. **Novos algoritmos para o problema de rotulação cartográfica de pontos**. 2003. Tese (Doutorado em Computação em Matemática Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2003.
- 13 YAMAMOTO M.; LORENA L.A.N. A constructive genetic approach to point-feature cartographic label placement. In: IBARAKI, T., NONOBE, K.; YAGIURA, M. (Ed.). **Metaheuristics: progress as real problem solvers**. Kluwer Academic Publishers, 2005. p. 285-300.
- 14 YAMAMOTO M.; CAMARA G.; LORENA L.A.N. Fast point-feature label placement algorithm for real time screen maps. In: Brazilian Symposium on Geoinformatics, 7. 2005, Campos do Jordão. **Anais...** Campos do Jordão: Geoinfo, 2005.