

Relaxação Lagrangeana com Formação de Clusters Aplicada a Rotulação Cartográfica de Pontos

Glaydston Mattos Ribeiro¹

Luiz Antonio Nogueira Lorena²

Laboratório Associado de Computação e Matemática Aplicada
Instituto Nacional de Pesquisas Espaciais
12.227-010, São José dos Campos – SP
{glaydston¹, lorena²}@lac.inpe.br

Resumo

Este trabalho apresenta uma nova maneira de relaxar um problema de otimização combinatória de modo a obter limites de melhor qualidade. Essa técnica foi aplicada com êxito a uma nova modelagem matemática proposta neste trabalho, para o Problema da Rotulação Cartográfica de Pontos (PRCP). Dado que um problema pode ser representado através de um grafo, essa relaxação consiste em dividir esse grafo em pequenos subproblemas (*clusters*) facilmente resolvidos. As arestas que conectam os clusters são então relaxadas no sentido Lagrangeano e sendo assim, com uso de um algoritmo de subgradiente, é possível resolver os problemas separadamente reagrupando-os após cada iteração. Esse método permitiu encontrar melhores resultados do que os relatados na literatura para o PRCP.

Palavras-chave: Relaxação Lagrangeana, Programação Inteira, Rotulação Cartográfica.

Abstract

This paper presents a new approach to relax a combinatorial optimization problem with the aim of obtain better bounds. This approach was applied with successful to a new mathematic model proposed in this work for the Point-Feature Cartographic Label Placement Problem (PFCLPP). Given that a problem can be represented by a graph, this relaxation divide the graph in small sub problems (clusters) easily solved. The edges which connect the clusters are relaxed in the Lagrangean way so, with the subgradient algorithm, it is possible to solve the separated problems, regrouping them after each iteration. This method allowed to find better results to PFCLPP than those reported in the literature.

Keywords: Lagrangean Relaxation, Integer Programming, Label Placement.

1. Introdução

O Problema da Rotulação Cartográfica de Pontos (PRCP) é muito semelhante ao Problema de Máximo Conjunto Independente de Vértices (PMCIIV). O problema consiste em rotular os pontos de um mapa de modo a evitar as sobreposições dos rótulos (Ver Figura 1). Dentro desse contexto, vários algoritmos com objetivos diferentes foram propostos na literatura, com isso, existem três abordagens distintas que muitas vezes são confundidas e analisadas em conjunto (Ribeiro e Lorena, 2004). O PRCP pode ser visto como um PMCIIV, como um Problema Máximo Número de Rótulos Sem Conflitos (PMNRSC) ou como um Problema de Minimização do Número de Conflitos (PMNC). As duas últimas abordagens são mais interessantes, sob o ponto de vista cartográfico, pois rotulam todos os pontos, sendo este trabalho direcionado à terceira abordagem.

Porém, para rotular os pontos em um mapa deve-se observar a padronização cartográfica (Christensen et al, 1995) que determina as possíveis localizações dos rótulos dos pontos. Ao se definir essas possíveis localizações, este problema pode ser formulado como um problema de otimização combinatória. A Figura 2 mostra um conjunto de 8 possíveis posições para o rótulo de um ponto, que

também são conhecidas por posições candidatas. O número presente em cada posição, determina a preferência cartográfica sendo a posição um, a de maior interesse para a cartografia.

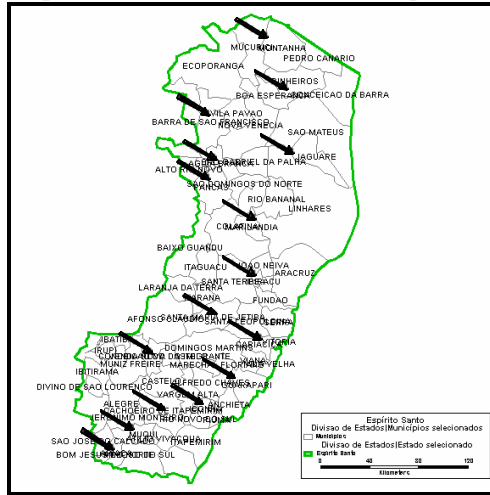


Figura 1. Rotulação Cartográfica de Pontos. As setas indicam alguns conflitos.

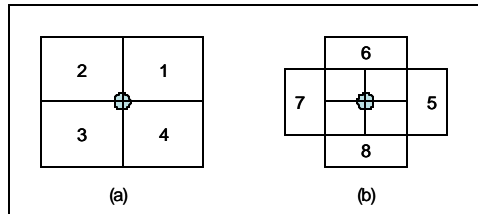


Figura 2. Conjunto de 8 posições candidatas para rotular um ponto (Christensen et al., 1995).

Considere um problema em que para cada ponto existem somente 4 posições candidatas, como mostra a Figura 3(a). O mesmo pode ser facilmente representado por um grafo de conflitos daí sua grande semelhança com um PMCIIV. Sendo N o número de pontos a serem rotulados e P o número de posições candidatas de cada ponto, seja $G=\{V,A\}$ o grafo de conflito, onde $V=\{v_1, v_2, \dots, v_{N \times P}\}$ representa o conjunto das posições candidatas (vértices) e $A=\{(v_i, v_j): i, j \in V, i \neq j\}$ as sobreposições (conflitos). A Figura 3(b) apresenta o grafo de conflito obtido para o problema da Figura 3(a) e a Figura 3(c) a solução ótima para esse problema. Normalmente na literatura para se determinar a qualidade de uma solução, deve-se obter o percentual de rótulos sem conflito. No caso da Figura 3(c) têm-se 100% de rótulos sem conflitos.

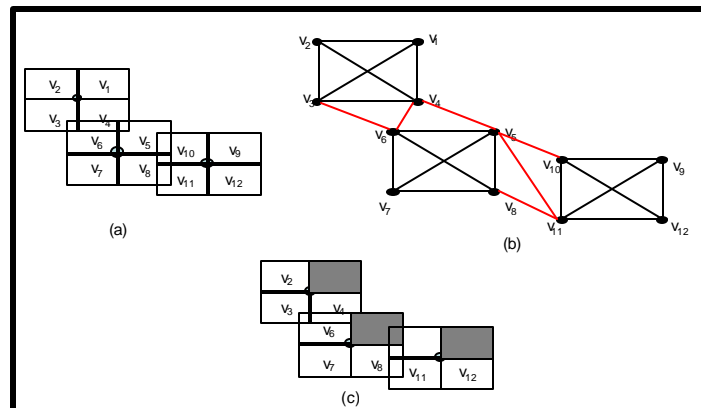


Figura 3. Representação em grafo do PRCP. (a) Problema, (b) Grafo relacionado e (c) Solução ótima.

Considerando o grafo acima, este trabalho tem como objetivo apresentar uma modelagem de programação inteira binária para o problema e aplicar uma relaxação Lagrangeana com formação de *clusters* (LAGCLUS). Os resultados encontrados foram melhores que os presentes na literatura.

O restante do artigo está organizando como segue: na próxima seção, é feita uma breve revisão bibliográfica sobre o PRCP, que é seguida pela modelagem matemática proposta. Na Seção 4 será mostrada a LAGCLUS, seguida pelos resultados computacionais e pelas conclusões.

2. Revisão Bibliográfica

Considerando o PRCP como um PMCIV, vários trabalhos propuseram diferentes algoritmos, assim como técnicas que permitem reduzir o número de restrições. Zoraster (1986, 1990 e 1991) formulou matematicamente o PRCP porém, para lidar com as restrições de conflito, criou posições candidatas fictícias de custo elevado, de tal modo que se nenhuma das reais posições candidatas pudessem ser usadas para posicionar um rótulo, a posição fictícia era então utilizada. Em seu trabalho o autor apresentou uma relaxação Lagrangeana obtendo alguns resultados para instâncias pequenas. Por outro lado, Strijk et al (2000) propuseram uma modelagem de programação inteira binária que usa restrições de corte para reduzir o número de restrições de conflito presentes no modelo. Essas técnicas de redução apareceram antes nos trabalhos de Moon e Chaudhry (1984) e Murray e Church (1996b), e consistem em utilizar conceitos de *cliques* para transformar algumas restrições em uma única, reduzindo assim, o número total de restrições do problema original. Os autores utilizaram uma relaxação de programação linear e aplicaram um algoritmo *Branch and Bound* para encontrar as soluções ótimas para os problemas testados. Devido às dificuldades encontradas, os autores ainda aplicaram e propuseram várias heurísticas, sendo elas: *Simulated Annealing*, Busca em Vizinhança Diversificada, *k-Opt* e Busca Tabu. Sendo a Busca Tabu o algoritmo que apresentou os melhores resultados para as instâncias por eles testadas.

Considerando a segunda abordagem ou seja, o PRCP como um PMNRSC, Christensen et al (1993; 1995) propuseram um método denominado Busca Exaustiva, que faz uma procura por soluções melhores alternando posições de rótulos previamente posicionados. Christensen et al (1995) também propuseram um algoritmo guloso com sucessivas otimizações locais e um algoritmo denominado *Discrete Gradient Descent* que considera as posições alternativas dos rótulos, porém, esse algoritmo, apesar de rápido, tem dificuldades para escapar de mínimos locais. Verner et al (1997) aplicou um algoritmo genético com máscara. Eles propuseram uma maneira de trabalhar com as máscaras de modo que, se um rótulo está em conflito, será permitido a troca de posições através de cruzamentos e mutações. Mais tarde, Yamamoto et al (2002) propuseram um algoritmo de Busca Tabu eficiente que forneceu muito bons resultados quando comparados com a literatura. Schreyer e Raidl (2002) utilizaram um Sistema de Colônia de Formigas (*Ant Colony System-ACS*) mas seus resultados não foram satisfatórios quando comparados com os obtidos por Yamamoto et al (2002). Yamamoto (2003) desenvolveu um algoritmo exato para o PMNRSC e aplicou o Algoritmo Genético Construtivo (AGC) proposto por Lorena e Furtado (2001). O algoritmo exato foi aplicado a instâncias de até 25 pontos, pois utiliza uma estrutura em árvore o que limitou a sua aplicação, em contrapartida, o AGC foi aplicado a instâncias com até 1000 pontos, fornecendo os melhores resultados até então. Vale ressaltar que esta abordagem não possui ainda uma modelagem matemática definida.

A terceira abordagem é mais recente e foi explorada primeiramente por Ribeiro e Lorena (2004). Ela considera a minimização do número de conflitos e, em alguns casos, as soluções encontradas são limitantes inferiores para a segunda abordagem. Considere a Figura 4, a solução mostrada em (a) e (b) são iguais pois apresentam o mesmo número de rótulos em conflitos (4), porém se considerarmos a representação em grafo mostrada na figura, a solução (b) é mais interessante pois apresenta um número de conflitos (arestas) menor que o da solução mostrada em (a). Porém, na optimalidade, a segunda abordagem é um limitante superior para esta terceira.

No entanto, essa abordagem no caso de conflitos inevitáveis, permite “espalhar” mais os rótulos, fornecendo uma melhor visualização do mapa. Ribeiro e Lorena (2004) propuseram uma modelagem e utilizaram várias relaxações, dentre elas a LAGCLUS, mas não obtiveram resultados melhores que os da literatura, exceto para problemas com até 500 pontos e 4 posições candidatas utilizando o CPLEX 7.5 (ILOG, 2001).

A modelagem matemática apresentada neste artigo foi inspirada no trabalho de Murray e Church (1996c). Isso permitiu reduzir bem o número de restrições geradas se comparada com a modelagem de Ribeiro e Lorena (2004).

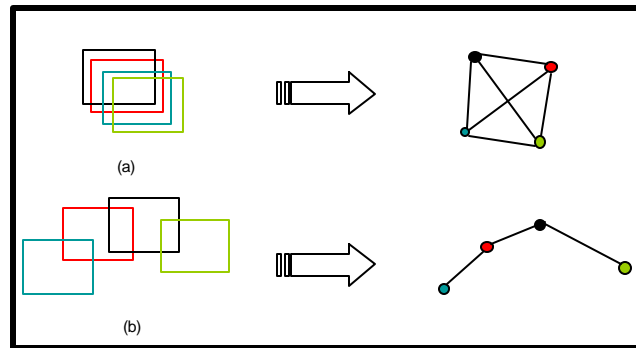


Figura 4. Comparação entre abordagens.

3. Modelagem Matemática Proposta para o PRCP

A modelagem matemática proposta, foi baseada, como dito anteriormente, no trabalho de Ribeiro e Lorena (2004) e no trabalho de Murray e Church (1996c), tendo como objetivo minimizar o número de conflitos. Formalmente, cada posição candidata é representada por uma variável binária $x_{i,j}$ tal que $i \in N$ e $j \in P_i$, sendo P_i o número de posições candidatas para o ponto i . Se $x_{i,j} = 1$ a posição candidata j do ponto i receberá o rótulo do ponto i . Além disso, cada posição candidata $x_{i,j}$ está associada a um custo $w_{i,j}$, o que permite priorizar algumas posições candidatas como mostrado na Figura 2.

Para cada ponto i apenas uma das posições candidatas deverá ser selecionada para receber o rótulo, ou seja, apenas uma das P_i posições receberá o valor 1. Com isso, essa restrição pode ser expressa por:

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

Por outro lado, ao se posicionar os rótulos, os conflitos devem ser evitados. Então, dado que para todo ponto i apenas uma das posições candidatas receberá o rótulo (restrições definidas em (1)), as restrições de conflitos podem ser representadas entre pontos ao invés de arestas como feito em Ribeiro e Lorena (2004). Com isso, como em Ribeiro e Lorena (2004), seja $S_{i,j}$ um conjunto de pares de índices $(k,t): k > i$ de posições candidatas $x_{k,t}$ conflitantes com $x_{i,j}$. Seja agora $C_{i,j}$ 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 $c \in C_{i,j}$. Assim, as restrições de conflitos definidas por Ribeiro e Lorena (2004) podem ser substituídas, sem falta de generalização, pelo seguinte conjunto de restrições:

$$\sum_{(k,t) \in S_{i,j}} x_{k,t} - \sum_{c \in C_{i,j}} y_{i,j,c} \leq |C_{i,j}| \quad \forall i = 1 \dots N \quad (2)$$

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

O problema então consiste em rotular todos os pontos (restrição (1)) evitando as sobreposições (restrição (2)). Para isso, a função objetivo deverá ser penalizada caso algum conflito apareça na solução, sendo assim, as variáveis de conflitos (y) devem aparecer na função objetivo que pode ser definida como:

$$v = \min \sum_{i=1}^N \sum_{j=1}^{P_i} w_{i,j} x_{i,j} + \sum_{c \in C_{i,j}} y_{i,j,c} \quad (3)$$

Dadas as definições acima, o Problema de Minimização do Número de Conflitos (PMNC) para o PRCP pode ser formalmente definido como segue:

$$v(\text{PMNC}) = \text{Min} \sum_{i=1}^N \sum_{j=1}^{P_i} w_{i,j} x_{i,j} + \sum_{c \in C_{i,j}} y_{i,j,c} \quad (4)$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \quad " i = 1 \dots N \quad (5)$$

$$\left| C_{i,j} \right| x_{i,j} + \sum_{(k,t) \in S_{i,j}} x_{k,t} - \sum_{c \in C_{i,j}} y_{i,j,c} \leq \left| C_{i,j} \right| \quad " i = 1 \dots N \quad (6)$$

$$" j = 1 \dots P_i$$

$$x_{i,j} \text{ e } y_{i,j,c} \in \{0,1\} \quad " i = 1 \dots N \quad (7)$$

$$" j = 1 \dots P_i$$

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

Essa formulação foi aplicada às instâncias propostas por Yamamoto e Lorena (2005) e estão disponíveis em <http://www.lac.inpe.br/~lorena/instancias.html>. A tabela abaixo mostra o número de restrições geradas pela modelagem acima e pela modelagem de Ribeiro e Lorena (2004), considerando que todos os pontos possuem 4 posições candidatas. A primeira coluna informa o número de pontos a serem rotulados, seguida pelo número de instâncias existentes, pelo número médio de restrições geradas pela modelagem de Ribeiro e Lorena (2004) e pelo número médio de restrições geradas pela nova modelagem proposta neste trabalho. Observa-se que essa segunda modelagem reduz consideravelmente o número de restrições.

Tabela 1. Número médio de restrições geradas.

Número de pontos	Número de instâncias	Ribeiro e Lorena (2004)	Modelagem Proposta
25	8	357	96
100	25	202	153
250	25	864	530
500	25	2909	1412
750	25	6181	2481
1000	25	10700	3643

4. Relaxação Lagrangeana com Formação de Clusters

Dado que um problema pode ser representado por um grafo de conflitos como mostrado na Seção 2, a relaxação Lagrangeana com formação de *clusters* (LAGCLUS) consiste em dividir o problema original em subproblemas menores com parte das restrições do problema original. As arestas que conectam os *clusters* são então relaxadas no sentido Lagrangeano. Isso permite com que os multiplicadores lagrangeanos sejam mais bem definidos, obtendo assim, limitantes melhores.

A Figura 5 mostra como essa técnica pode ser aplicado ao PRCP para a formulação definida na Seção 3. Como as restrições de conflito são baseadas em pontos, o grafo (b) do problema mostrado em (a), é então transformado em um grafo de pontos que é dividido em clusters, através de alguma técnica de particionamento de grafos, como mostrado em (c). A partir do grafo particionado, refaz-se o problema o original como mostrado em (d). Assim ao final, as arestas que conectam os clusters (e) são então relaxadas no sentido Lagrangeano e com isso, obtém-se subproblemas menores que podem ser resolvidos separadamente, como mostrado em (f).

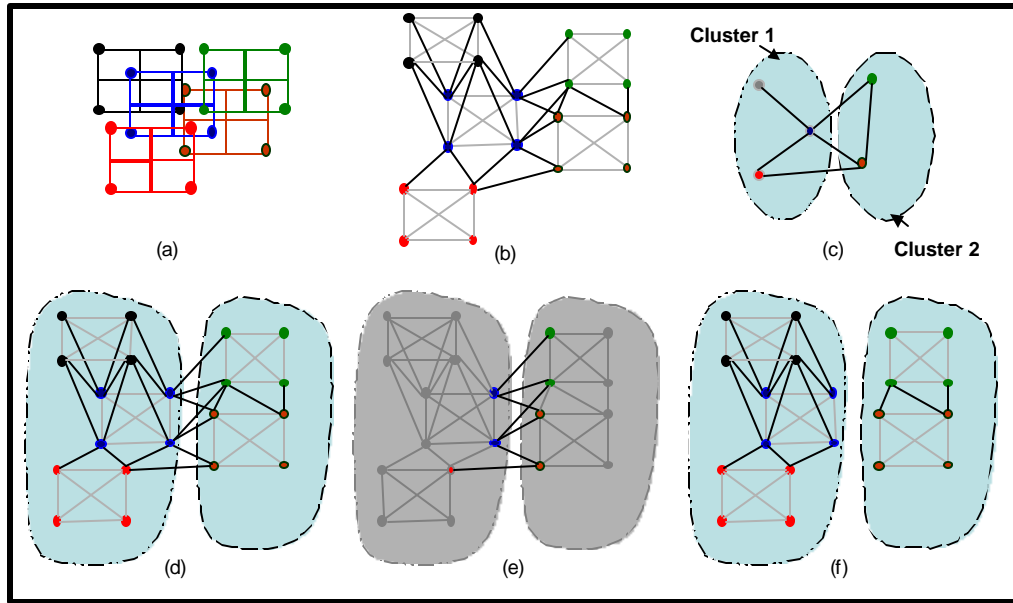


Figura 5. Relaxação Lagrangeana com formação dos clusters

O particionamento do grafo deve ser feito de tal forma que o número de arestas que conectam os subgrafos seja minimizado. Neste trabalho foi utilizado o METIS (Karyps e Kumar, 1998) que é uma heurística bem conhecida para particionamento de grafos. Com isso, de forma resumida a LAGCLUS para o PMNC consiste em (por questões de notação, considere P sendo PMNC):

- i. Aplicar uma heurística de particionamento para dividir o grafo G em m partes, formando m clusters. O problema P pode ser escrito através de uma função objetivo definida em (4) sujeita a (5) - (7), sendo o conjunto de restrições de conflitos (6) dividido em dois grupos: um com as restrições de conflito intra-clusters, e outro formado pelas restrições de conflitos que corresponde as arestas que conectam os clusters (arestas de corte).
- ii. Multiplicadores distintos maiores ou iguais a zero, relaxam, no sentido Lagrangeano, o grupo das restrições de conflito que correspondem às arestas de corte.
- iii. A relaxação Lagrangeana é então decomposta em m sub-problems e resolvida.

Ao resolver os subproblemas gerados, as soluções encontradas são soluções factíveis para o P , pois a restrição (5) não é violada. Foi utilizado o algoritmo de Subgradientes descrito por Narciso e Lorena (1999), que aproxima as soluções de um problema no sentido euclidiano de distância (Parker e Rardin, 1988). O controle utilizado para o parâmetro p foi o mesmo proposto por Held e Karp (1971). Inicia-se com 2 e se durante 15 iterações o valor do limite superior (ub) não decrescer, p é dividido pela metade. Sendo lb o limite inferior e g^l os subgradientes encontrados para os multiplicadores lagrangeanos l , as condições de parada consideradas foram:

- 1) $p \leq 0,005$
- 2) $(ub - lb) < 1$
- 3) $\|g^l\|^2 = 0$

Durante cada passo do algoritmo de Subgradientes, foi utilizado a heurística de melhoria (HM) (descrita a seguir) proposta por Ribeiro e Lorena (2004), que aproveita a solução relaxada e tenta obter uma solução primal melhor.

Heurística de Melhoria - HM

1. Para cada elemento da Solução factível, armazene em um vetor de conflito o número de conflitos para cada posição.
2. Para $i=1$ até o tamanho do vetor de conflitos;
3. Se Vetor de conflitos[i] $\neq 0$
4. Procurar dentre as possíveis posições candidatas j , a que apresenta o menor número de conflitos com a solução viável atual.
5. Se houver alguma posição candidata j com número de conflitos inferior a Vetor de conflitos[i], trocar Solução factível[i] por posição candidata j .
6. Senão não alterar Solução factível[i];
7. Fim do Para.

5. Resultados Computacionais

A LAGCLUS foi aplicada aos problemas testes propostos por Yamamoto e Lorena (2005), com 4 posições candidatas para cada ponto, sem preferência cartográfica, ou seja, $w_{i,j}=1 \quad i=1..N$ e $j=1..4$ e, os subproblemas gerados foram resolvidos utilizando o CPLEX 7.5 (ILOG, 2001). A Tabela 2 mostra os resultados médios obtidos com o CPLEX utilizando a modelagem proposta, para as instâncias testes. O CPLEX foi executado até resolver a instância ou parar por falta de memória. A máquina utilizada foi um Pentium IV com 512MB de memória RAM. A primeira coluna (Problema) exibe o número de pontos a serem rotulados, seguida pelo limite inferior (Limite Inferior) e superior (Limite Superior), e pelo $GAP=(Limite Superior-Limite Inferior)/Limite Superior$ que fornece, em porcentagem, o *gap* encontrado entre os dois limitantes. Na quarta coluna são mostrados os tempos computacionais utilizados, seguidos pelos números de rótulos em conflito e pelos percentuais de rótulos livres para cada problema.

Para os problemas com até 500 pontos, o CPLEX resolveu todos em um tempo computacional muito baixo (menor que 0,74s), entretanto para problemas maiores, ele parou muitas vezes por falta de memória. Para problemas com 750 pontos, em 12 das 25 instâncias, ele encontrou a optimalidade, já para 1000 pontos, nenhuma solução ótima foi encontrada.

Tabela 2. Resultados médios obtidos com o CPLEX (ILOG, 2001).

CPLEX						
Problema	Limite Inferior	Limite Superior	GAP	Tempo (s)	Rótulos em conflitos	Percentual de rótulos livres
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%

A Tabela 3 apresenta os resultados médios obtidos com a aplicação da LAGCLUS que foi implementada em C++. Nesta tabela, além dos campos mostrados na Tabela 2, está o número de clusters usados em cada problema. Pode-se verificar facilmente que os *gaps* de dualidade encontrados são menores que os obtidos pelo CPLEX, com exceção dos problemas com 25 pontos, entretanto vale ressaltar que essas instâncias foram preparadas por Yamamoto e Lorena (2002) para terem o mesmo grau de dificuldade das instâncias com 1000 pontos (Yamamoto e Lorena, 2005).

Para os problemas maiores (750 e 1000 pontos), a qualidade das soluções obtidas (ver Percentual de rótulos livres) foi bem superior às obtidas pelo CPLEX, em um tempo computacional menor.

Para as instâncias com 750 pontos, a LAGCLUS encontrou a optimalidade em 21 das 25 instâncias. A Figura 6 mostra o comportamento da LAGCLUS para a instância 3 dos problemas com 750 pontos, percebe-se que os limites se aproximam rapidamente. Para problemas com 1000 pontos, assim como aconteceu com o CPLEX, nenhuma solução ótima foi encontrada, no entanto o *gap* de dualidade foi 2 vezes menor.

Tabela 3. Resultados médios obtidos com a LAGCLUS.

LAGCLUS							
Problema	Número de Clusters	Limite Inferior	Limite Superior	GAP	Tempo (s)	Rótulos em conflitos	Percentual de rótulos livres
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%

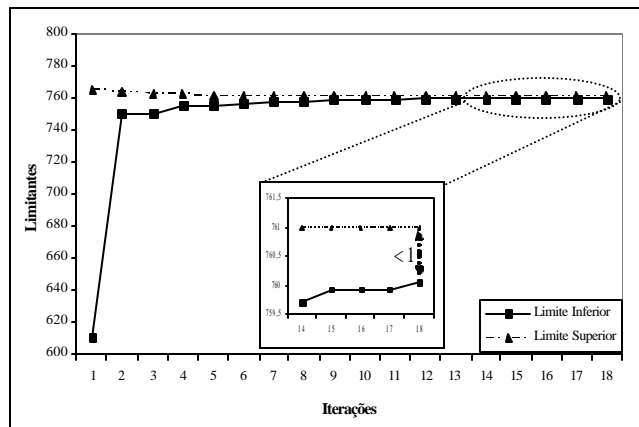


Figura 6. Evolução da LAGCLUS para instância 3 com 750 pontos.

Como nas instâncias com 1000 pontos não foi possível encontrar com a LAGCLUS a solução ótima em nenhuma delas, o número de *clusters* foi reduzido para 20 pois, como o número de arestas particionadas é reduzido, os subproblemas gerados são mais fortes gerando assim limitantes melhores que podem levar a optimalidade. Para mostrar o que acontece também quando se aumenta o número de *clusters*, foi realizado o mesmo experimento com 30 *clusters*. Os resultados estão mostrados na Tabela 4.

Percebe-se claramente que, com a redução do número de *clusters* as soluções obtidas foram melhores (ver GAP e Percentual de rótulos livres) no entanto, o tempo computacional aumentou e nenhuma solução ótima foi encontrada. Por outro lado, ao aumentar o número de *clusters* as soluções encontradas foram piores que as obtidas com 20 e 25 *clusters*, porém o tempo computacional reduziu.

Tabela 4. Resultados médios obtidos com a LAGCLUS para as instâncias com 1000 pontos.

Número de Clusters	Limite Inferior	Limite Superior	GAP	Tempo (s)	Rótulos em conflitos	Percentual de rótulos livres
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%

A Tabela 5 vem sendo muito utilizada para comparar os algoritmos propostos, no entanto, pode-se observar que eles vêm sendo comparados como se tivessem os mesmos objetivos, não respeitando suas respectivas abordagens como mostrado anteriormente. Os algoritmos que consideram o máximo conjunto independente de vértices, não rótula pontos quando o conflito é inevitável, os algoritmos que buscam o maior número de rótulos livres, deixam o mapa muitas vezes, com áreas ilegíveis (como mostrou a Figura 4); e por último, os algoritmos que procuram tornar o mapa mais legível

minimizando os conflitos, buscam “espalhar” mais os rótulos o que é diferente da segunda abordagem. Ribeiro e Lorena (2004) em seu trabalho, também apontam essas diferenças. Não foi possível fazer comparações entre os tempos computacionais dos algoritmos pois os computadores utilizados foram todos diferentes. Contudo, percebe-se claramente que a LAGCLUS produziu melhores resultados do que todos os algoritmos existentes na literatura para o PRCP. Para uma descrição mais detalhada dos mesmos, ver Verner et al (1997), Yamamoto et al (2002) e Yamamoto e Lorena (2005).

Tabela 5. Comparação com

Algoritmo	Percentual de rótulos livres				
	Problema				
	100	250	500	750	1000
LAGCLUS	100,00	100,00	99,67	97,65	91,42/90,98/90,64
PMNC Exato – CPLEX	100,00	100,00	99,67	97,62	90,29
AGC – Melhor	100,00	100,00	99,60	97,10	90,70
AGC – Média	100,00	100,00	99,60	96,80	90,40
Busca Tabu	100,00	100,00	99,30	96,80	90,00
GA com Máscara	100,00	99,98	98,79	95,99	88,96
GA sem máscara	100,00	98,40	92,59	82,38	65,70
<i>Simulated Annealing</i>	100,00	99,90	98,30	92,30	82,09
Zoraster	100,00	99,79	96,21	79,78	53,06
Hirsh	100,00	99,58	95,70	82,04	60,24
<i>3-opt Gradient Descent</i>	100,00	99,76	97,34	89,44	77,83
<i>2-opt Gradient Descent</i>	100,00	99,36	95,62	85,60	73,37
<i>Gradient Descent</i>	98,64	95,47	86,46	72,40	58,29
Algoritmo Guloso	95,12	88,82	75,15	58,57	43,41
Busca exaustiva	84,56	65,63	44,06	29,06	19,53

Fonte: Adaptado de Verner et al. (1997), Yamamoto et al (2002) e Yamamoto e Lorena (2005)

6. Conclusões

Este trabalho apresentou uma nova modelagem para o problema da rotulação cartográfica de pontos e a relaxação Lagrangeana com formação de *clusters*. Essa nova modelagem reduziu consideravelmente o número de restrições geradas e ainda permitiu encontrar as soluções ótimas de problemas que até então, eram desconhecidas. Por outro lado, a nova forma de relaxar um problema que pode ser representado através de um grafo, forneceu resultados muito bons, os melhores, com *gaps* de dualidade pequenos, para todos os problemas da literatura.

Como trabalhos futuros, seria interessante fechar os *gaps* de dualidade ainda existentes através de um *Branch and Bound*, com isso, certamente os resultados serão ainda melhores. A LAGCLUS poderá ser aplicada a outros problemas que podem ser representados por grafo.

Agradecimentos

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPQ – pelo apoio financeiro dado ao desenvolvimento deste trabalho.

Referências

1. Christensen, J., Marks, J., e Shieber, S. (1993) **Placing Text Labels on Maps and Diagrams**. London. Academic Press.
2. Christensen, J., Marks, J., e Shieber, S. (1995) An Empirical Study of Algorithms for Point-Feature Label Placement. **ACM Transactions on Graphics**, v 14, n 3, p 203-232.
3. Held, M. and Karp, R. M. (1971). The Traveling Salesman Problem and Minimum Spanning Trees: Part II. **Mathematical Programming**, 1: 6-25.
4. ILOG, (2001), ILOG CPLEX. Copyright 2001 by ILOG.
5. Karypis, G. e Kumar, V., (1998) Multilevel *k-way* Partitioning Scheme for Irregular Graphs. **Journal of Parallel and Distributed Computing** 48, p 96-129.

6. Lorena, L. A. N. e Furtado, J. C., (2001) Constructive Genetic Algorithm for Clustering Problems. **Evolutionary Computation** 9, v 3, p 309-327.
7. Moon, I. D., e Chaudhry, S. (1984), An Analysis of Network Location Problems with Distance Constraints. **Management Science** 30, 290-307.
8. Murray, A. T., e Church, R. L., (1996b) Constructing and Selecting Adjacency Constraints. **INFOR** p 232-248.
9. Murray, A. T., e Church, R. L., (1996c) Facets for Node Packing. **European Journal of Operational Research** 101, p 598-608.
10. Narciso, M. G., e Lorena, L. A. N., (1999) Lagrangean/Surrogate Relaxation for Generalized Assignment Problems. **European Journal of Operational Research** 114, p 165-177.
11. Parker, R. G. e Rardin, R. L., (1988), **Discrete Optimization**. Academic Press, New York, USA.
12. Ribeiro, G. M., e Lorena, L. A. N., (2004), Modelagem Matemática e Relaxações Lagrangeana e Lagrangeana/Surrogate para o Problema de Rotulação Cartográfica de Pontos. **SBPO XXXVI** – São João Del Rei – MG.
13. Schreyer, M., e Raidl, G. R., (2002), Letting Ants Labeling Point Features. **In Proc. of the 2002 IEEE Congress on Evolutionary Computation at the IEEE World Congress on Computational Intelligence**, p 1564-1569.
14. Strijk, T., Verweij, B. e Aardal, K., (2000) 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>.
15. Verner, O. V., Wainwright R. L., e Schoenfeld, D. A. (1997), Placing Text Labels on Maps and Diagrams Using Genetic Algorithms with Masking. **INFORMS Journal on Computing**, v 9, p 266-275.
16. Yamamoto, M., Camara, G., e 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 Publisher. Ntherlands, 6:1, p 77-90.
17. Yamamoto, M., e 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).
18. Yamamoto, M., (2003), **Novos algoritmos para o problema de rotulação cartográfica de pontos**. Tese de doutorado em Computação em Matemática Aplicada, INPE.
19. Zoraster, S. (1991), Expert Systems and the Map Label Placement Problem. **Cartographica**, v. 28, n 1, p. 1-9.
20. Zoraster, S. (1986), Integer Programming Applied to the Map Label Placement Problem. **Cartographica** , v. 23, n. 3, p. 16-27.
21. Zoraster, S. (1990), The Solution of Large 0-1 Integer Programming Problems Encountered in Automated Cartography. **Operations Research** , v. 38, n 5, p 752-759.