

# MODELAGEM MATEMÁTICA E RELAXAÇÕES LAGRANGIANA E LAGRANGEANA/SURROGATE PARA O PROBLEMA DE ROTULAÇÃO CARTOGRÁFICA DE PONTOS

**Glaydston Mattos Ribeiro**

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

**Luiz Antonio Nogueira Lorena**

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

## Resumo

O problema de rotulação cartográfica tem como objetivo dar uma maior legibilidade a um mapa. Muitas pesquisas o consideram como um problema de otimização combinatória. Algumas abordagens distintas foram propostas de modo a obter o maior número de pontos rotulados considerando que sobreposições não podem ocorrer, ou ainda, rotular todos os pontos e obter o maior número de rótulos sem sobreposições. Este trabalho tem como objetivo abordar o problema de uma outra forma, minimizando o número de sobreposições existentes em uma rotulação de todos os pontos de um mapa. Uma formulação matemática de programação linear inteira binária é apresentada seguida de algumas relaxações: a relaxação lagrangeana, a lagrangeana/surrogate e outra relaxação lagrangeana constituída a partir da decomposição do problema original com formação de agrupamentos (clusters).

**Palavras-chave:** Rotulação, Heurísticas, Programação Inteira.

## Abstract

The point-feature cartographic label placement problem objectives to give more legibility to an automatic map creation, placing point labels in clear and legible positions. Many researches consider distinct combinatorial optimization formulations, such as to obtain the maximum number of labeled points considering that overlaps in labels can not happen, or to obtain the maximum number of point labels without overlaps considering that all points must be labeled. This paper considers the problem in another way, minimizing the number of existing overlaps in a labeling of all points on a map. A mathematical formulation of binary integer linear programming is presented, followed by some relaxations: the lagrangean relaxation, the lagrangean/surrogate and a lagrangean relaxation formed after the decomposition of the original problem in clusters.

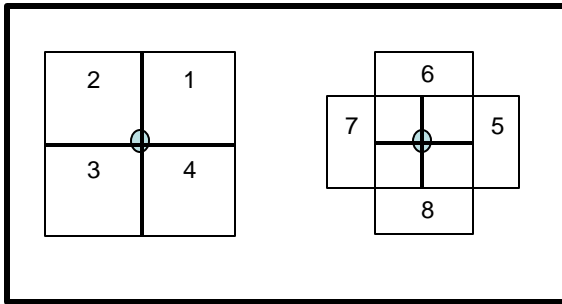
**Keywords:** Labeling, Heuristics, Integer Programming.

## 1. Introdução

A rotulação cartográfica de pontos pode ser considerada um problema de otimização combinatória. O problema consiste em posicionar rótulos de pontos em determinadas posições, de tal forma a atender a um conjunto de restrições, minimizando ou maximizando uma função objetivo. Para cada ponto existe uma lista de possíveis posições (posições candidatas) para que um determinado rótulo seja colocado. Essa lista é escolhida de acordo com uma padronização cartográfica (Christensen et al, 1995) que tem como objetivo priorizar determinadas posições. A Figura 1 mostra um conjunto de 8 posições candidatas para um determinado ponto no qual, os números indicam a preferência

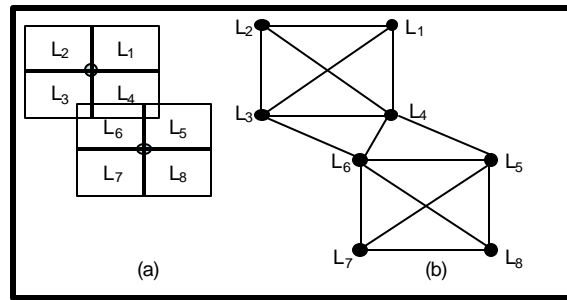
cartográfica: quanto menor for o número, maior é a preferência cartográfica. Por outro lado, ao se colocar rótulos em determinadas posições candidatas, conflitos podem surgir, gerando problemas na visualização do mapa. Assim, devido a esses conflitos, um problema de rotulação cartográfica (PRC) de  $N$  pontos pode ser representado através de um grafo  $G=\{V,A\}$ , onde  $V=\{v_1, v_2, \dots, v_N\}$  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 2 (b) mostra o grafo que representa o problema mostrado na Figura 2 (a) que tem dois pontos, cada um com 4 posições candidatas para seus respectivos rótulos. Verifica-se que a posição candidata  $L_3$  possui um conflito com a posição  $L_6$  e  $L_4$  possui dois: um com  $L_5$  e outro com  $L_6$ . A partir dessa representação, surgiram abordagens distintas para o problema. Esse problema pode ser considerado como sendo um problema de Máximo Conjunto Independente de Vértices (Zoraster, 1990; Strijk et al., 2000) e como um problema de Maximização do Número de Rótulos sem Conflito.



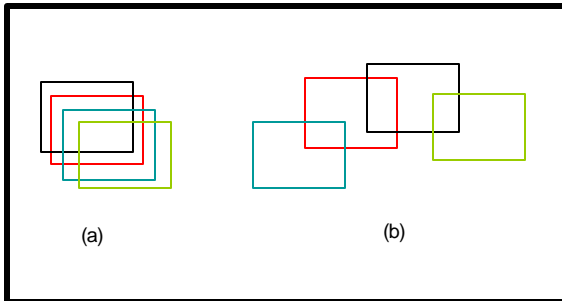
**Figura -1** Conjunto de 8 posições candidatas para rotular um ponto.

FONTE: Christensen et al (1995, p. 205)



**Figura -2** Posições candidatas, conflitos e o grafo relacionado.

FONTE: Yamamoto e Lorena (2003)



**Figura -3** Soluções possíveis para um problema com 4 pontos.

A diferença entre essas duas abordagens está no número de pontos rotulados. Na primeira abordagem nem todos os pontos são rotulados, pontos com sobreposições inevitáveis são desconsiderados, ou seja, não são rotulados. Já na segunda, todos os pontos são rotulados objetivando obter o maior número possível de rótulos sem conflito.

A segunda abordagem é mais relevante que a primeira, sob o ponto de vista cartográfico, pois possibilita rotular todos os pontos, porém, em determinadas situações, podem ocorrer mapas “sobrecarregados” visualmente. A Figura 3

mostra duas possíveis soluções para um problema com quatro pontos. Considerando a segunda abordagem, as duas soluções são iguais pois apresentam todos os rótulos em conflito, por outro lado, a solução (b) é mais atraente (menos “sobrecarregada”) que a solução (a).

Assim, considerando a questão do mapa “sobrecarregado” visualmente, este trabalho tem o objetivo de propor uma terceira abordagem para o PRC, contribuindo com o desenvolvimento de um modelo de programação linear inteira para o Problema da Minimização do Número de Conflitos (PMNC), e também, baseado nessa modelagem proposta, algumas relaxações.

O restante do artigo está organizado como segue: na próxima seção, é feita uma breve revisão bibliográfica sobre o PRC, que é seguida pela modelagem matemática proposta. Na Seção 4 serão mostradas as relaxações desenvolvidas, seguidas pelos resultados computacionais e pelas conclusões.

## 2. Revisão Bibliográfica

Considerando o PRC como sendo um Problema de Máximo Conjunto Independente de Vértices (PMCIIV), vários pesquisadores propuseram diferentes algoritmos, assim como novas técnicas que permitem reduzir o número de restrições geradas. O PMCIIV tem várias aplicações em diferentes

campos como em seqüenciamento de DNA (Joseph et al, 1992), localização de defesas militares (Chaudhry et al, 1986), modelos de alocação (Gerrard e Church, 1996), *Ant-Covering* (Murray e Church, 1996a), planejamento florestal (Murray e Church, 1996b) e Church et al (1998), Cortes e Empacotamento (Beasley, 1985) e Carregamento de Paletes (Oliveira, 2003).

Zoraster (1986, 1990 e 1991) formulou matematicamente o PRC 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 utilizada. Ele propôs também uma relaxação Lagrangeana para o problema e mostrou alguns resultados computacionais para algumas instâncias, não obtendo sucesso em instâncias de grande porte. Strijk et al (2000) propõe um algoritmo que utiliza Busca Tabu. Yamamoto (2003) descreve esse algoritmo e verifica seu desempenho comparando-o com resultados da literatura.

Contudo, devido às várias aplicações desse tipo de problema, algumas pesquisas foram realizadas de modo a reduzir o número de restrições geradas, considerando-as de uma outra forma mais sucinta. Murray e Church (1996c) afirmam existirem duas opções de pesquisa para PMCIIV: o desenvolvimento de técnicas alternativas para solucioná-los ou desenvolvimento de representações alternativas para as restrições. Em seu trabalho, os autores exploram a segunda alternativa no qual os mesmos apresentam uma nova estrutura híbrida de restrição que permitem identificar a solução ótima para problemas de tamanho moderado. Os autores afirmam que uma formulação particular para esse tipo de problema pode ter um grande impacto resolvendo com sucesso vários tipos de problemas de programação inteira. Essa estrutura híbrida combina dois tipos de restrições: uma baseada no Clique Maximal e a outra no Problema de Cobertura de Vértices proposto por Moon e Chaudhry (1984). Os autores aplicaram com sucesso o método em grafos com até 150 vértices.

Por outro lado, muitos trabalhos têm sido feitos considerando a segunda abordagem, o Problema de Maximizar o Número de Rótulos sem Conflito (PMNRSC). Christensen et al (1993; 1995) propôs um método denominado Busca Exaustiva, que faz uma procura por soluções melhores alternando posições de rótulos previamente posicionados. Christensen et al (1995) também propôs um algoritmo guloso com sucessivas otimizações locais e um algoritmo denominado *Discrete Gradient Descent* que considera posições alternativas dos rótulos, porém, esse algoritmo tem uma dificuldade de escapar de mínimos locais. Hirsch (1982) desenvolveu um algoritmo dinâmico de repulsão de rótulos, onde rótulos em conflito são movidos na tentativa de eliminar o conflito. Verner et al (1997) aplicou um algoritmo genético com máscara. Ele propôs uma maneira de trabalhar com as máscaras de modo a permitir que se um rótulo está em conflito, será permitido a troca de posições através de cruzamentos e mutações.

Yamamoto et al (2002) propôs um algoritmo de Busca Tabu eficiente que forneceu resultados muito bons quando comparados com a literatura. Mais tarde, 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, quando comparados com a literatura.

### 3. Modelagem Matemática Proposta para o PRC

A formulação matemática proposta para o PRC tem como objetivo, como dito anteriormente, minimizar o número de conflitos, assim, para cada ponto  $i$  existe um número fixo  $P_i$  de posições candidatas. Cada posição candidata é representada por uma variável binária  $x_{i,j}$  onde  $i = 1..N$ , sendo  $N$  o número de pontos a serem rotulados; e  $j = 1..P_i$ . Se  $x_{i,j} = 1$  a posição candidata  $j$  do ponto  $i$  será alocada (receberá o rótulo do ponto  $i$ ), caso contrário,  $x_{i,j} = 0$ . Além disso, a cada posição candidata é associado um custo (uma preferência cartográfica) representado por  $w_{i,j}$ .

Cada posição candidata  $x_{i,j}$  possui um conjunto  $S_{i,j}$  de pares de índices de posições candidatas conflitantes com  $x_{i,j}$ .  $S_{i,j}$  é o conjunto de pares de índices  $(k,t)$  de posições candidatas  $x_{k,t}$  que são conflitantes com  $x_{i,j}$ . Para todo  $(k,t) \in S_{i,j}$ , onde  $k \in \{1..N\}$ :  $k > i$  e  $t \in \{1..P_k\}$ , existe uma variável binária  $y_{i,j,k,t}$  que representa o conflito, uma aresta no grafo  $G$ .

Assim, considerando as informações acima, a função objetivo do Problema de Minimização do Número de Conflitos (PMNC) pode ser representada por:

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

Porém, para cada ponto  $i$  apenas uma posição candidata deve ser selecionada, ou seja, apenas umas das posições candidatas receberá o valor 1. Essa restrição pode ser escrita como:

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

Com relação aos conflitos, ao se posicionar um rótulo em uma determinada posição candidata, deve-se levar em consideração as sobreposições existentes, assim, um novo conjunto de restrições se faz necessário. Esse conjunto leva em consideração cada posição  $x_{i,j}$ , suas respectivas posições de conflito  $x_{k,t}$  e uma variável de conflito  $y_{i,j,k,t}$ . Sendo assim, têm-se:

$$x_{i,j} + x_{k,t} - y_{i,j,k,t} \leq 1 \quad " \quad i = 1 \dots N \\ " \quad j = 1 \dots P_i \\ (k,t) \in S_{i,j} \quad (3)$$

Além das restrições definidas acima, deve-se garantir que todas as variáveis de decisão sejam binárias. Com isso, se faz necessário à inclusão das seguintes restrições:

$$x_{i,j}, x_{k,t} \text{ e } y_{i,j,k,t} \in \{0,1\} \quad " \quad i = 1 \dots N \\ " \quad j = 1 \dots P_i \\ (k,t) \in S_{i,j} \quad (4)$$

Observa-se que as variáveis de conflito aparecem somando na função objetivo, que, ao ser minimizada, deverá procurar eliminá-las ou minimizá-las (caso a eliminação não seja possível). Essa formulação, é parecida com a proposta por Zoraster (1990), porém ela permite posicionar todos os rótulos minimizando o número de conflitos.

Por outro lado, essa formulação quando aplicada a problemas com até 500 pontos forneceu, com auxílio do CPLEX 7.5 (ILOG, 2001), as soluções ótimas em poucos segundos. Em contrapartida, para instâncias maiores com 750 e 1000 pontos resolvidas por Yamamoto et al (2002) e Yamamoto (2003), os resultados ótimos não foram obtidos. Assim, algumas relaxações foram propostas de modo a permitir resolver o PMNC.

#### 4. Relaxações Propostas para o PMNC

##### 4.1 Relaxação Lagrangeana

O modelo definido em (1)-(4) foi resolvido utilizando-se uma relaxação Lagrangeana, com a otimização de subgradiente e uma heurística específica. Fisher (1981, 1985) afirma que quando esse três elementos são utilizados de tal forma a produzir um algoritmo de otimização, dá-se o nome de Heurística Lagrangeana. A restrição a ser relaxada no sentido Lagrangeano, semelhante ao que Zoraster (1990) fez, será a de conflitos (3). Então, para um dado vetor de multiplicadores  $\lambda \in R_+^C$  onde  $C$  é o número de restrições de conflito, que pode ser obtido da seguinte maneira:

$$C = \sum_{i=1}^N \sum_{j=1}^{P_i} |S_{i,j}|, \quad (5)$$

uma relaxação Lagrangeana para o PMNC (por questões de notação, considere  $P$  sendo PMNC) pode ser definida como sendo:

$$v(L_1 P) = \text{Min} \sum_{i=1}^N \sum_{j=1}^{P_i} \xi_{i,j} w_{i,j} x_{i,j} + \sum_{(k,t) \in S_{i,j}} \lambda_{i,j,k,t} y_{i,j,k,t} + \sum_{i=1}^N \sum_{j=1}^{P_i} \lambda_{i,j,k,t} (x_{i,j} + x_{k,t} - y_{i,j,k,t}) - \sum_{i=1}^N \sum_{j=1}^{P_i} \lambda_{i,j,k,t} \quad (6)$$

Sujeito a (2) e (4).

O valor ótimo  $v(L_1 P)$  é menor ou igual a  $v(P)$  e resulta da solução do problema *dual* lagrangeano  $\text{Max}_{\lambda \geq 0} \{v(L_1 P)\}$ . Para isso, foi considerado o algoritmo de Subgradientes descrito por Lorena e Narciso (1999), que aproxima as soluções de um problema no sentido euclidiano de distância (Parker e Rardin, 1988). Em uma iteração  $l$ , os subgradientes são definidos como  $g^l = x_{i,j}^l + x_{k,t}^l - y_{i,j,k,t}^l - 1$ ,  $i \in \{1, \dots, N\}$ ,  $j \in \{1, \dots, P_i\}$  e  $(k,t) \in S_{i,j}$ , tal que  $(x^l, y^l)$  está na solução ótima de  $(L^l P)$ . O método de subgradientes atualiza o multiplicador  $\lambda_l$  fazendo  $\lambda_{l+1} = \lambda_l + \mathbf{q}_l g^l$ , onde  $\mathbf{q}_l$  é o tamanho do passo calculado como  $\mathbf{q}_l = \frac{\mathbf{p}(ub_l - v(L^l P))}{\|g^l\|^2}$ , sendo  $ub_l$  o valor de uma solução melhorada

(descrita a seguir) para  $P$  encontrada na iteração  $l$ . O controle utilizado para o parâmetro  $\mathbf{p}$  é o mesmo proposto por Held e Karp (1971). Inicia-se com 2 e se durante 15 iterações o valor do limite superior ( $ub_l$ ) não decrescer,  $\mathbf{p}$  é dividido pela metade. As condições de parada consideradas são:

- 1)  $\mathbf{p} \leq 0.005$ ;
- 2)  $(ub_l - v(L^l P)) < 1$ ;
- 3)  $\|g^l\|^2 = 0$ ;

Uma solução relaxada  $Sol_{rel} = (x^l, y^l)$  de  $(L^l P)$ , composta das variáveis de posições candidatas  $(x_{i,j}^l)$  e de variáveis de conflito  $(y_{i,j,k,t}^l)$ , também é uma solução factível para  $P$  pois, a restrição de conflito (3) é relaxada mas o conjunto de restrições (2) é respeitado. Essa solução factível é melhorada utilizando a seguinte heurística de busca local.

#### 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.

#### 4.2 Relaxação Lagrangeana/Surrogate

A heurística Lagrangeana/Surrogate (*LagSur*) (Narciso e Lorena, 1999) também foi utilizada no problema  $P$  para relaxar o conjunto de restrições de conflitos (3). Primeiramente é feita a relaxação no sentido *Surrogate* como descrito por Glover (1968) sobre (3), e em seguida no sentido Lagrangeano. Assim, para um dado vetor de multiplicadores  $\lambda \in R_+^C$ , onde  $C$  é o número de restrições de conflito

definido em (5), uma relaxação *Surrogate* para  $P$  pode ser encontrada, em seguida, ao se relaxar a restrição *surrogate* encontrada utilizando um multiplicador dual  $t \geq 0$ , pode-se obter a seguinte relaxação *LagSur* para o problema  $P$ :

$$v(L_t SP_1^I) = \text{Min} \sum_{i=1}^N \sum_{j=1}^{P_i} \lambda_{i,j} w_{i,j} x_{i,j} + \sum_{(k,t) \in S_{i,j}} \alpha_{i,j,k,t} y_{i,j,k,t} \quad (7)$$

$$+ t \sum_{i=1}^N \sum_{j=1}^{P_i} \lambda_{i,j} (x_{i,j} + x_{k,t} - y_{i,j,k,t}) - \sum_{i=1}^N \sum_{j=1}^{P_i} \lambda_{i,j} \quad (8)$$

Sujeito a (2) e (4).

Uma característica interessante dessa relaxação é que para  $t = I$  obtém-se a relaxação Lagrangeana mostrada anteriormente. Para o vetor  $I$  fixo, o melhor valor para  $t$  pode ser calculado resolvendo-se o seguinte problema dual:

$$v(D_t^I)_I = \text{Max}_{t \geq 0} v(L_t SP_1^I) \quad (8)$$

O valor ótimo  $v(D_t^I)_I$  fornece um limitante melhor que o obtido pela relaxação lagrangeana (Senne e Lorena, 2000). Senne e Lorena (2000) descrevem um algoritmo de busca dicotômica que aproxima o melhor valor de  $t$ . Sendo  $t^*$  esse melhor valor, se para um dado número de iterações do algoritmo de subgradiente o valor de  $t^*$  repete, então esse valor é mantido fixo até o final do procedimento e a busca não é mais executada. Para a interação  $l$ , os multiplicadores são atualizados como  $I_{l+1} = I_l + q_l g^{I_l}$ , onde  $q_l$  é o tamanho do passo calculado por  $q_l = \frac{p(ub_l - v(L_{t^*} SP_1^{I_l}))}{\|g^{I_l}\|^2}$ , e

$ub_l$  é uma solução factível encontrada pelo heurística HM.

A relaxação *LagSur* também foi aplicada sobre o conjunto de restrições (2). Da mesma forma, dado um vetor de multiplicadores  $I \in R^N$  onde  $N$  é o número de pontos, obtém-se uma relaxação *Surrogate* para  $P$  que pode ser relaxada novamente no sentido lagrangeano utilizando um multiplicador dual  $t$  irrestrito, obtendo assim uma relaxação *LagSur* que pode ser expressa como:

$$v(L_t SP_2^I) = \text{Min} \sum_{i=1}^N \sum_{j=1}^{P_i} \lambda_{i,j} w_{i,j} x_{i,j} + \sum_{(k,t) \in S_{i,j}} \alpha_{i,j,k,t} y_{i,j,k,t} \quad (9)$$

$$+ t \sum_{i=1}^N \sum_{j=1}^{P_i} \lambda_{i,j} x_{i,j} - \sum_{i=1}^N \lambda_i$$

Sujeito a (3) e (4)

Agora, os subgradientes são  $g^{I_l} = \sum_{j=1}^{P_i} \lambda_{i,j} x_{i,j} - 1 \quad i=1, \dots, N$ , e o método de subgradiente atualiza os multiplicadores  $I_l$  como mostrado anteriormente,  $I_{l+1} = I_l + q_l g^{I_l}$ , onde  $q_l$  é o tamanho do passo calculado por  $q_l = \frac{p(ub_l - v(L_{t^*} SP_2^{I_l}))}{\|g^{I_l}\|^2}$ , e  $ub_l$  é uma solução factível encontrada pelo heurística de construção (HC), descrita abaixo.

A heurística de construção constrói uma solução factível a partir da solução relaxada  $Sol_{rel} = (x^I, y^I)$  de  $(L_{t^*} SP_2^I)$ , encontrando posições candidatas (restrição 2) com o menor número de conflitos com

a solução factível que está sendo construída. Se algum ponto  $i$  não está rotulado, a heurística procura dentre todas as possíveis posições candidatas  $j$ , aquela que apresenta o menor conjunto de conflitos definidos em  $S_{i,j}$ . A solução factível obtida é então melhorada usando a heurística HM.

#### Heurística de Construção - HC

1. Preencher Vetor solução factível com zeros;
2. Para  $i=1$  até  $N$ 
  3. Procurar em solução relaxada todas as posições diferentes de zero para o ponto  $i$ ;
  4. Selecionar para solução factível na posição  $i$ , a posição candidata  $j$  com o menor número de conflitos com os elementos da solução factível corrente. Em caso de igualdade, selecionar aquela que possui o menor conjunto  $S_{i,j}$ .
  5. Se nenhuma posição candidata  $j$  para o ponto  $i$  está na solução relaxada, escolher a posição candidata que apresentar o menor conjunto  $S_{i,j}$ .
6. Fim do Para.

### 4.3 A Relaxação Lagrangeana com Formação de Clusters

Considere a Figura 4 abaixo. O grafo obtido para o problema mostrado é particionado em duas partes, formando dois agrupamentos (*clusters*). Ao se dividir, dois subproblemas são formados, menores do que o problema original. Isso permite que cada um seja resolvido rapidamente de forma independente. Porém, essa divisão faz com que algumas restrições não sejam mais consideradas, que são as arestas de ligação entre *clusters*. Assim, baseado nessa idéia e considerando o problema original  $P$ , propõe-se uma nova relaxação no sentido Lagrangeano feita da seguinte maneira:

- i. Aplicar uma heurística de particionamento para dividir  $G$  em  $m$  partes, formando  $m$  *clusters*. O problema  $P$  pode ser escrito através de uma função objetivo definida em (1) sujeita a (2) - (4), onde, o conjunto de restrições de conflitos (3) é 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. Dois multiplicadores distintos relaxam no sentido Lagrangeano, as restrições definidas em (2) e o grupo das restrições de conflito que correspondem as arestas de corte.
- iii. A relaxação Lagrangeana é decomposta em  $m$  sub-problems e resolvida. De agora em diante, essa relaxação será chamada de *LagClus*.

Ao se relaxar o conjunto de restrições definido em (2), novamente uma solução relaxada pode não ser uma solução factível para  $P$ , assim, as heurísticas HC e HM são usadas para obter soluções factíveis.

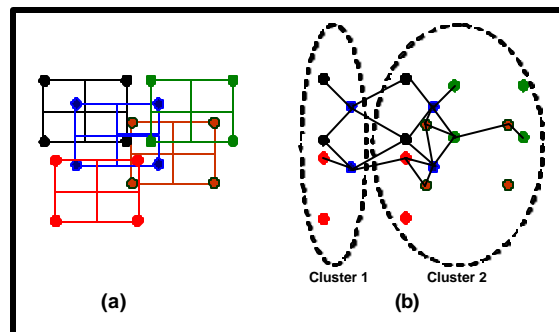


Figura -4 Formação de Clusters.

A divisão do grafo  $G$  foi realizada utilizando o METIS (Karyps e Kumar, 1998) que é uma heurística bem conhecida para o problema de particionamento de grafos. Dado um grafo  $G=\{V,A\}$  e um número pré-definido  $m$  de *clusters*, o METIS divide o grafo  $G$  em  $m$  agrupamentos com o objetivo de minimizar o número de arestas que tem terminações em *clusters* diferentes.

Recentemente Hicks et al (2004) aplicaram esta técnica a um problema de conjunto independente máximo de peso (*Maximum Weight Independent Set Problem*), conseguindo bons resultados.

## 5. Resultados Computacionais

Os testes computacionais foram feitos sobre as instâncias propostas por Yamamoto e Lorena (2003) disponíveis em <http://www.lac.inpe.br/~lorena/instancias.html>. O programa, em C++, e os testes, foram feitos em um computador com um processador Pentium 2.66 GHz e 512 MB de memória RAM. Como realizado em Zoraster (1990), Yamamoto e Lorena (2002) e Yamamoto (2003), para todas 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.

A Tabela 1 mostra os resultados médios (cada problema tem 25 instâncias) obtidos após aplicação da relaxação lagrangeana e *LagSur* para o conjunto de restrições (3). A coluna *Problema* diz respeito ao problema em questão (número de pontos a serem rotulados), a coluna *Solução Ótima* a solução ótima obtida com o CPLEX, seguida por *Lower Bound* o limite dual, *Upper Bound* o melhor

limite superior, *gap\_ub*, calculado como sendo  $Gap\_ub = \frac{Upper\ bound - Solução\ Ótima}{Solução\ Ótima} * 100$ ,

fornecendo, em porcentagem, o *gap* encontrado entre o *Upper Bound* e a solução ótima do problema,

depois, *gap\_lb*, calculado como sendo  $Gap\_lb = \frac{Solução\ Ótima - Lower\ bound}{Solução\ Ótima} * 100$  fornecendo,

em porcentagem, o *gap* encontrado entre o *Lower Bound* encontrado e a solução ótima do problema, *iter* representa o número de iterações necessárias pelo algoritmo de subgradientes até atingir uma das condições de parada, *Tempo* o tempo necessário em segundos para executar o algoritmo e a última coluna (*Número*) diz respeito ao número de rótulos obtidos com conflito ao final do algoritmo, assim como o percentual de rótulos sem conflitos (%Total). Esse parâmetro foi obtido para posterior comparação com resultados da literatura, apesar dos problemas serem diferentes. Verifica-se ainda, na Tabela 1, algumas subdivisões aplicadas às colunas. *Lagrangeana* diz respeito à aplicação da relaxação lagrangeana como descrito na Seção 4.1, *Lagrangeana/Surrogate* considera a aplicação da relaxação *LagSur* para o conjunto de restrições definido em (3) com valores pré-definidos do multiplicador lagrangeano *t* e *Aleatória* diz respeito a um conjunto de 100 soluções geradas aleatoriamente, no qual para cada uma foi aplicada a heurística de melhoria descrita na Seção 4.1. O multiplicador *t* foi fixado na relaxação *LagSur* pois, ao se realizar a busca dicotômica pelo melhor *t* descrita por Senne e Lorena (2000), verificou-se que o mesmo estava tendendo a zero, o que era de se esperar pois toda informação relativa aos conflitos foi relaxada.

A solução ótima não pode ser obtida para os problemas com 750 e 1000 pontos devido à explosão combinatória. Os problemas de 100 e 250 pontos, são problemas simples onde uma solução sem conflito é obtida rapidamente em poucas iterações, com isso, o algoritmo é interrompido pois a solução ótima foi encontrada. Isso não permite que o limite inferior (*Lower Bound*) se aproxime da solução ótima, conseqüentemente os cálculos relacionados aos *Gaps* não foram realizados sendo substituídos por NC (Não Calculado). Esse problema também surgiu ao se resolver às instâncias com 500 pontos. Assim, os resultados mostrados nas Tabelas 1, 2 e 4 correspondem aos problemas em que o processo de otimização foi realizado até atingir uma das três condições de parada descritas na Seção 4.1. Outro ponto importante que foi considerado em todas as tabelas de resultados diz respeito às soluções ótimas dos problemas maiores (750 e 1000 pontos). Essas soluções não puderam ser encontradas, como dito anteriormente, porém para os cálculos foi considerada a melhor solução viável conhecida. Apesar de não ser a solução ótima, esta próxima. Isso permite estimar o quão próximo ou distante as soluções encontradas estão da melhor solução viável conhecida.

Voltando à Tabela 1, na relaxação lagrangeana verifica-se que o *gap* do *upper bound* variou de 0,00% a 4,14%, considerando a relaxação *LagSur* esse *gap*, no melhor caso, variou de 0,00% a 2,20%. Por outro lado, os resultados encontrados com as soluções aleatórias obtidos após a aplicação da heurística de melhoria, foram melhores variando de 0,00% a 2,06%. Isso demonstra uma questão interessante com relação às relaxações realizadas, pois ao se realizar a relaxação do conjunto de restrições de conflitos, toda a informação referente às sobreposições não são levadas em consideração, não permitindo ao algoritmo sair de mínimos locais. Isso pode ser comprovado quando se analisa o comportamento do limite *dual*, observa-se que em nenhum momento, o *dual* ultrapassou o limite



imposto pelo tamanho do problema. Isso demonstra que essa relaxação não é muito boa, podendo ser considerada como sendo fraca para esse problema. Os tempos obtidos foram bem razoáveis, variando de 0,00 segundos para problemas menores, a 19,60 segundos para problemas maiores (1000 pontos).

Os resultados da relaxação *LagSur* para o conjunto de restrições definido em (2) estão mostrados na Tabela 2. Apesar dessa relaxação ser mais forte, o resultado não foi interessante. Somente o limite inferior foi melhorado, mas o tempo consumido foi maior, o limite superior não foi melhorado e os problemas maiores (de 500, 750 e 1000 pontos) não puderam ser resolvidos devido a questões de tempo. Por exemplo, a instância 7 dos problemas de 25 pontos, consumiu 444 segundos para ser concluída.

A Tabela 3 se refere à relaxação Lagrangeana com formação de *clusters*, nela a primeira coluna mostra a instância, seguida do número de posições possíveis para rotular cada ponto, depois o número de vértices gerados no grafo, o número de *clusters* utilizado e por último, o número possível de vértices considerado em cada cluster. A Tabela 4 exibe os resultados encontrados após a aplicação da *LagClus*. Verifica-se que os resultados encontrados são de melhor qualidade do que todas as relaxações realizadas anteriormente. Os problemas maiores, 750 e 1000 pontos, foram resolvidos, em média, em 337,80 e 817,00 segundos, respectivamente. Os *gaps* superiores variaram de 0,00% a 1,48%, ou seja, estão muito próximo da solução ótima. Já os *gaps* inferiores variaram de 0,76% a 9,27%, resultados melhores do que as outras relaxações consideradas. Observa-se que de um modo geral, esta foi a que apresentou os melhores resultados, apesar de consumir um tempo maior. A Tabela 5 faz uma comparação das porcentagens médias encontradas de rótulos sem conflito usando a formulação PMNC e obtidas das relaxações, com os melhores resultados da literatura. No melhor resultado das relaxações do PMNC, depois de viabilizado, os rótulos em conflito foram contabilizados. Como esperado, os resultados foram inferiores porém para os problemas com 500 pontos os resultados foram melhores que o encontrado com o AGC.

## 6. Conclusões

Este trabalho apresentou uma nova formulação para o problema de rotulação cartográfica de pontos, sob o enfoque de uma formulação matemática diferente das encontradas na literatura. Essa abordagem considera a minimização do número de conflitos, o que permite, como mostrado anteriormente, uma maior legibilidade do mapa. Baseado na formulação realizada, três relaxações foram propostas, obtendo melhores resultados com relaxação lagrangeana com formação de *clusters*. Verificou-se que em muitos casos os resultados encontrados estavam iguais ou muito próximos da solução ótima. Com relação à relaxação lagrangeana aplicada ao conjunto de restrições de conflito, verificou-se que as informações referentes aos conflitos, por estarem relaxadas, influenciaram no processo de solução, não permitindo ao algoritmo sair de mínimos locais. Essa relaxação foi inspirada no trabalho de Zoraster (1990), assim, é provável que ele tenha encontrado esse mesmo problema, comprovando que se trata de uma relaxação fraca. Analisando de forma conjunta todos os resultados apresentados nas Tabelas 1, 2 e 4, os *gaps* encontrados foram satisfatórios, variaram de 0,00% a 3,21% em relação ao limite superior e de 0,00% a 9,72% em relação ao limite inferior. Os tempos computacionais também foram razoáveis, tendo como valor médio máximo 817 segundos para problemas de grande porte. É interessante notar que mesmo com objetivos diferentes, a média das soluções exatas dos problemas com 500 pontos forneceu um valor melhor do que o melhor resultado encontrado na literatura. Acredita-se que esse trabalho contribui para o problema de rotulação cartográfica e que ainda, pode ajudar na solução de outros problemas relacionados que podem ser formulados como um Problema de Máximo Conjunto Independente de Vértices.

## 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. (1995) An Empirical Study of Algorithms for Point-Feature Label Placement. **ACM Transactions on Graphics**, v 14, n 3, p 203-232.
2. Christensen, J., Marks, J., e Shieber, S. (1993) **Placing Text Labels on Maps and Diagrams**. London. Academic Press.
3. Zoraster, S. (1991), Expert Systems and the Map Label Placement Problem. **Cartographica**, v. 28, n 1, p. 1-9.
4. Zoraster, S. (1986), Integer Programming Applied to the Map Label Placement Problem. **Cartographica**, v. 23, n. 3, p. 16-27.
5. 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.
6. 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](http://ftp.cs.uu.nl/pub/RUU/CStechreps/CS-2000/2000-22.ps.gz).
7. Yamamoto, M., Camara, G., e Lorena, L.A.N, (2002), 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, 6:1, p 77-90.
8. Yamamoto, M., e Lorena, L.A.N, (2003), A Constructive Genetic Approach to Point-Feature Cartographic Label Placement. **In the Fifth Metaheuristics International Conference**.
9. Joseph, D., Meidanis, J., e Tiwari, P. (1992), Determining DNA Sequence Similarity Using Maximal Independent Set Algorithms for Interval Graphs. In: O. Nurmi and E. Ukkonen, Algorithm Theory, **SWAT** 92.
10. Chaudry, S., McCormick, S. e Moon, I. D., (1986), Locating Independent Facilities with Maximum Weight: Greedy Heuristics. **Omega** 14, p 383-389.
11. Gerrard, R. A. e Church, R. L., (1996) Closest Assignment Constraints and Location Models: Properties and Structure. **Location Science** 4, p 251-270.
12. Murray, A. T., e Church, R. L., (1996a) Solving the Anti-Covering Location Problem Using Lagrangian Relaxation. **Computers Operations Research**, v 24, n 2, p 127-140.
13. Murray, A. T., e Church, R. L., (1996b) Constructing and Selecting Adjacency Constraints. **INFOR** p 232-248.
14. Murray, A. T., e Church, R. L., (1996c) Facets for Node Packing. **European Journal of Operational Research** 101, p 598-608.
15. Church, R. L., Murray, A. T. e Weintraub, A., (1998) Locational Issues in Forest Management. **Location Science** 6, p 137-153.
16. Beasley, J., (1985) An Exact Two-Dimensional Non Guillotine Cutting Tree Search Procedure. **Operations Research**, v 33, p 49-64.
17. Oliveira, L. K., (2003) **Métodos Exatos Baseados em Relaxações Lagrangiana e Surrogate para o Problema de Carregamento de Paletes do Produtor**. Monografia de Exame de Qualificação, UFSCar.
18. Yamamoto, M., (2003), **Novos Algoritmos para Rotulação Cartográfica de Pontos**. Tese de Doutorado em Computação Aplicada, INPE.
19. Moon, I. D., e Chaudry, S. (1984), An Analysis of Network Location Problems with Distance Constraints. **Management Science** 30, 290-307.
20. Hirsch, S. A., (1982), An Algorithm for Automatic name Placement around Point Data. **American Cartographer**, v 9, n 1, p 5-17.
21. Verner, O. V., Wainwright R. L., e Schoenefeld, D. A. (1997), Placing Text Labels on Maps and Diagrams Using Genetic Algorithms with Masking. **INFORMS Journal on Computing**, v 9, p 266-275.
22. Fisher, M. L., (1981). The Lagrangian Relaxation Method for Solving Integer Programming Problem. **Management Science** 27, p 1-18.
23. Fisher, M. L., (1985). An Applications Oriented Guide to Lagrangian Relaxation. **Interfaces** 15, p 10-21.
24. Parker, R. G. e Rardin, R. L., (1988), **Discrete Optimization**. Academic Press, New York, USA.
25. Glover, F. (1968), Surrogate Constraints. **Operations Research** 16, p 741-749.
26. Senne, E. L. F., e Lorena, L., A., N, (2000), Lagrangian/Surrogate Heuristics for p-Median Problems. **In Computing Tools for Modeling, Optimization and Simulation**. Kluwer Academic Publishers, New York, p 115-130.
27. Hicks, I. V., Warren, J. S., Warrier, D. e Wilhelm W. E., (2004), A Branch-and-Price Approach for the Maximum Weight Independent Set Problem. (Working paper). (On line) <http://ie.tamu.edu/People/faculty/Hicks/>.
28. Karypis, G. e Kumar, V., (1998) Multilevel *k*-way Partitioning Scheme for Irregular Graphs. **Journal of Parallel and Distributed Computing** 48, p 96-129.
29. Lorena, L. A. N. e Furtado, J. C., (2001) Constructive Genetic Algorithm for Clustering Problems. **Evolutionary Computation** 9, v 3, p 309-327.
30. 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.
31. ILOG, (2001), ILOG CPLEX. Copyright 2001 by ILOG.

**Tabela 1** – Resultados médios para 25 instâncias de cada problema, obtidos após aplicação das duas primeiras relaxações.

Problema	Solução Ótima	Lower bound				Upper bound					Gap ub (%)				
		$(L_1P)$	$(L_tSP_1^I)$			$(L_1P)$	$(L_tSP_1^I)$			Aleatória	$(L_1P)$	$(L_tSP_1^I)$			Aleatória
			t=0,25	t=0,50	t=0,75		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	28,88	28,38	28,25	28,63	28,00	4,14	2,33	1,82	3,21	0,93
100,00	100,00	NC	NC	NC	NC	100,00	100,00	100,00	100,00	100,00	0,00	0,00	0,00	0,00	0,00
250,00	250,00	NC	NC	NC	NC	250,00	250,00	250,04	250,00	250,00	0,00	0,00	0,02	0,00	0,00
500,00	500,84	498,20	493,46*	496,91*	499,45	503,92	502,88	503,16	503,80	502,48	0,61	0,41	0,46	0,59	0,33
750,00	758**	749,98	743,42	748,74	749,98	774,44	771,80	772,68	774,44	771,04	2,17	1,82	1,94	2,17	1,72
1000,00	1056**	999,93	997,04	999,93	999,95	1086,44	1075,64	1079,24	1086,40	1077,80	2,88	1,86	2,20	2,88	2,06

**Tabela 1** – Resultados médios obtidos após aplicação da primeira heurística lagrangeana (Continuação)

Problema	Gap lb (%)					Iter					Solução Ótima	Tempo (s)				
	$(L_1P)$	$(L_tSP_1^I)$			Aleatória	$(L_1P)$	$(L_tSP_1^I)$			Aleatória		$(L_1P)$	$(L_tSP_1^I)$			Aleatória
		t=0,25	t=0,50	t=0,75			t=0,25	t=0,50	t=0,75				t=0,25	t=0,50	t=0,75	
25,00	9,72	9,87	9,72	9,72	9,72	146,63	148,88	148,38	146,63	101,00	1,60	0,25	0,13	0,13	0,13	0,25
100,00	NC	NC	NC	NC	0,00	1,16	1,16	1,16	1,16	2,48	0,02	0,00	0,00	0,00	0,04	0,00
250,00	NC	NC	NC	NC	0,00	2,48	2,36	7,72	2,68	3,00	0,06	0,00	0,00	0,04	0,00	0,00
500,00	0,53	1,48*	0,80*	0,28	0,17	146,00	140,24	134,48	146,20	88,16	3,12	2,76	2,64	2,64	2,80	2,44
750,00	1,06	1,92	1,22	1,06	1,06	146,00	146,00	146,00	146,00	101,00	8,00**	8,48	8,44	8,40	8,56	9,56
1000,00	5,31	5,58	5,31	5,31	5,30	146,00	146,00	146,00	146,00	101,00	18,00*	19,32	19,60	19,52	19,08	23,96

**Tabela 1** – Resultados médios obtidos após aplicação da primeira heurística lagrangeana (Continuação).

Problema	Número e Percentual de Rótulos sem Conflito											
	Solução Ótima		$(L_1P)$		$(L_tSP_1^I)$						Aleatória	
	Número	%Total	Número	%Total	t=0,25	%Total	t=0,50	%Total	t=0,75	%Total	Número	%Total
25,00	4,88	80,50	7,13	71,50	6,50	74,00	5,88	76,50	6,63	73,50	5,38	78,50
100,00	0,00	100,00	0,00	100,00	0,00	100,00	0,00	100,00	0,00	100,00	0,00	100,00
250,00	0,00	100,00	0,00	100,00	0,00	100,00	0,08	99,97	0,00	100,00	0,00	100,00
500,00	1,64	99,67	7,76	98,45	5,72	98,86	6,32	98,74	7,52	98,50	4,96	99,01
750,00	16,00**	97,87**	46,08	93,86	42,04	94,39	43,00	94,27	46,08	93,86	41,00	94,53
1000,00	111,00**	88,90**	159,40	84,06	142,56	85,74	148,24	85,18	159,44	84,06	147,20	85,28

\*Média entre os resultados que forneceram soluções com conflitos.

\*\*Melhor solução encontrada com as relaxações  $(L_1P)$  e  $(L_tSP_1^I)$ , de todas as instâncias.

**Tabela 2** – Resultados médios para 25 instâncias de cada problema, obtidos para a relaxação *LagSur* aplicada sobre o conjunto de restrições (2).

Problema	Solução Ótima	Lower bound	Upper bound	Gap ub (%)	Gap lb (%)	Iter	Tempo (s)		Número de Rótulos com Conflito e Percentual de Rótulos sem Conflito			
		$(L_iSP_2^I)$	$(L_iSP_2^I)$	$(L_iSP_2^I)$	$(L_iSP_2^I)$	$(L_iSP_2^I)$	Solução Ótima	$(L_iSP_2^I)$	Solução Ótima		$(L_iSP_2^I)$	
									Number	%Total	Number	%Total
25	27,75	25,13	28,38	2,29	9,27	151,63	1,60	104,63	4,88	80,50	6,00	76,00
100	100,00	NC	100,00	0,00	NC	1,00	0,02	0,16	0,00	100,00	0,00	100,00
250	250,00	NC	250,00	0,00	NC	1,48	0,06	0,92	0,00	100,00	0,00	100,00

Obs: Para os problemas com 500, 750 e 1000 pontos as soluções ótimas não foram encontradas devido ao alto tempo necessário para solucionar o problema.

**Tabela 3** – Número de *Clusters* considerado para cada problema.

Problema	Número de Possíveis Posições para o Rótulo	Número de Vértices	Número de Clusters	Número de Possíveis Vértices em cada Cluster.
25	4	100	2	50
100	4	400	4	100
250	4	1000	10	100
500	4	2000	20	100
750	4	3000	25	120
1000	4	4000	60	66,67

**Tabela 5** – Comparação dos resultados médios das porcentagens de rótulos sem conflitos com os melhores resultados encontrados na literatura.

Algoritmo	25(%)	100(%)	250(%)	500(%)	750(%)	1000(%)
Exato (Yamamoto, 2003)	83,50	NC	NC	NC	NC	NC
AGC (Yamamoto, 2003)	80,75	100,00	100,00	99,60	97,10	90,70
Solução Exata do PMNC	80,50	100,00	100,00	99,67	NC	NC
$(L_iP)$	71,54	100,00	100,00	98,45	93,86	84,06
$(L_iSP_1^I)^*$	76,50	100,00	100,00	98,86	94,39	85,74
$(L_iSP_2^I)$	76,00	100,00	100,00	NC	NC	NC
LagClus	77,50	100,00	100,00	99,38	95,53	86,47

\*Melhores resultados : NC = Não Calculado

**Tabela 4** – Resultados médios para 25 instâncias de cada problema, obtidos com a relaxação *LagClus*.

Instância	Solução Ótima	Lower bound	Upper bound	Gap ub (%)	Gap lb (%)	Iter	Time (s)		Número de Rótulos com Conflito e Percentual de Rótulos sem Conflito			
		Lagrangeana	Lagrangeana	Lagrangeana	Lagrangeana	Lagrangeana	Solução Ótima	Lagrangeana	Solução Ótima		Lagrangeana	
									Número	%Total	Número	%Total
25	27,75	25,13	27,88	0,46	9,27	148,50	1,60	23,88	4,88	80,50	5,63	77,50
100	100,00	NC	100,00	0,00	NC	1,00	0,02	0,16	0,00	100,00	0,00	100,00
250	250,00	NC	250,00	0,00	NC	7,12	0,06	2,36	0,00	100,00	0,00	100,00
500	500,84	497,30*	501,52	0,14	0,76*	103,16	3,12	82,72	1,64	99,67	3,08	99,38
750	757,00**	749,41	767,08	1,33	1,00	145,28	0,00	337,80	14,00**	98,13**	33,56	95,53
1000	1055,00**	1002,11	1070,60	1,48	5,01	145,96	0,00	817,00	107,00**	89,30**	135,32	86,47

\*Média entre os resultados que forneceram soluções com conflitos.

\*\*Melhor solução encontrada de todas as instâncias.