MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2015/05.15.19.04-TDI

DISPERSÃO DISCRETA E DECOMPOSIÇÃO LAGRANGEANA DESBALANCEADA PARA O PROBLEMA DE ROTULAÇÃO CARTOGRÁFICA DE PONTOS

Sóstenes Pereira Gomes

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Luiz Antonio Nogueira Lorena, e Glaydston Mattos Ribeiro, aprovada em 18 de maio de 2015.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34P/3JG3TF8>

INPE
São José dos Campos
2015

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GB)
Serviço de Informação e Documentação (SID)
Caixa Postal 515 - CEP 12.245-970
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/6921
Fax: (012) 3208-6919
E-mail: pubtc@sid.inpe.br

COMISSÃO DO CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELECTUAL DO INPE (DE/DIR-544):
Presidente:
Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)
Membros:
Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)
Dr. Amauri Silva Montes - Coordenação Engenharia e Tecnologia Espaciais (ETE)
Dr. André de Castro Milone - Coordenação Ciências Espaciais e Atmosféricas (CEA)
Dr. Joaquim José Barroso de Castro - Centro de Tecnologias Espaciais (CTE)
Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos (CPT)
Drª Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação
Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)
BIBLIOTECA DIGITAL:
Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)
Clayton Martins Pereira - Serviço de Informação e Documentação (SID)
REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:
Simone Angélica Del Ducca Barbedo - Serviço de Informação e Documentação (SID)
Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)
EDITORAÇÃO ELETRÔNICA:
Marcelo de Castro Pazos - Serviço de Informação e Documentação (SID)
André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)

MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2015/05.15.19.04-TDI

DISPERSÃO DISCRETA E DECOMPOSIÇÃO LAGRANGEANA DESBALANCEADA PARA O PROBLEMA DE ROTULAÇÃO CARTOGRÁFICA DE PONTOS

Sóstenes Pereira Gomes

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Luiz Antonio Nogueira Lorena, e Glaydston Mattos Ribeiro, aprovada em 18 de maio de 2015.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34P/3JG3TF8>

INPE
São José dos Campos
2015

Dados Internacionais de Catalogação na Publicação (CIP)

Gomes, Sóstenes Pereira.

G585d Dispersão discreta e decomposição lagrangeana desbalanceada para o problema de rotulação cartográfica de pontos / Sóstenes Pereira Gomes. – São José dos Campos : INPE, 2015.

xxii + 78 p. ; (sid.inpe.br/mtc-m21b/2015/05.15.19.04-TDI)

Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2015.

Orientadores : Drs. Luiz Antonio Nogueira Lorena, e Glaydston Mattos Ribeiro.

1. Problema de rotulação cartográfica de pontos. 2. Otimização combinatória. 3. Dispersão discreta. I.Título.

CDU 519.85

Esta obra foi licenciada sob uma Licença Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada.

This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License.

Aprovado pela Banca Examinadora em cumprimento a requisito exigido para a obtenção do Título de **Doutor(a)** em **Computação Aplicada**

Dr. Nandamudi Lankalapalli Vijaykumar
Presidente / INPE /São José dos Campos - SP

Dr. Luiz Antonio Nogueira Lorena
Orientador(a) /INPE/ /São José dos Campos - SP

Dr. Glaydston Mattos Ribeiro
Orientador(a) /UFRJ/ Rio de Janeiro - RJ

Dr. Solon Venâncio de Carvalho
Membro da Banca /INPE/ São José dos Campos -SP

Dr. José Carlos Becceneri
Membro da Banca /INPE/ São José dos Campos -SP

Dr. Luiz Leduíno de Salles Neto
Convidado(a) /UNIFESP/ São José dos Campos - SP

Dr. Fernando Augusto Silva Marins
Convidado(a) /UNESP/Guaratinguetá- SP

Este trabalho foi aprovado por:
( ) maioria simples.
(X) unanimidade.

Título: "Dispersão discreta e decomposição lagrangeana desbalanceada para o problema de rotulação cartográfica de pontos."

Aluno (a): Sóstenes Pereira Gomes

São José dos Campos, 18 de Maio de 2015.

A meus pais Neusa e Idalme, e a meus irmãos Halley e Dâmaris.

AGRADECIMENTOS

Agradeço a Deus pela vida e oportunidades concedidas.

A meus pais, irmãos e familiares pelo amor incondicional e apoio.

Aos Professores Drs. Luiz Antônio Nogueira Lorena e Glaydston Mattos Ribeiro, pela paciência, orientação, ideias e conhecimentos compartilhados que permitiram a realização deste trabalho.

Ao INPE pela oportunidade de completar meus estudos.

Aos professores da CAP, com os quais muito aprendi durante este período.

À CAPES pelo auxílio financeiro.

A todos os funcionários do INPE, em especial aos das secretarias da CAP e do LAC.

Ao Geraldo Regis Mauri por disponibilizar o programa da Decomposição Lagrangeana.

Por fim, agradeço a todos os amigos e colegas de pós-graduação do INPE pelo companheirismo, em especial à Ligia Correa, que me apoiou incondicionalmente e esteve presente nos momentos mais difíceis.

RESUMO

Este trabalho aborda o Problema de Rotulação Cartográfica de Pontos (PRCP), que é um problema de otimização combinatória, demonstrado na literatura ser NP-difícil. Considera-se que quando todos os pontos devem ser rotulados e sobreposições de rótulos são inevitáveis, o mapa pode ser mais legível se os rótulos em conflito são posicionados de maneira dispersiva, isto é, se os rótulos são posicionados o mais distante possível. Neste contexto, é apresentada uma nova abordagem para o problema, denominada Dispersão Discreta, já utilizada em Problemas de Localização de Facilidades em diversos trabalhos da literatura. Com esta nova abordagem, o PRCP foi formulado como um Problema de programação inteira mista, de maneira a considerar as distâncias entre posições candidatas. Um algoritmo genético construtivo também foi desenvolvido, para obter resultados em instâncias mais complexas. Por fim, é proposta uma Decomposição Lagrangeana desbalanceada, que permitiu obter a solução ótima de diversas instâncias do PRCP, além de provar a otimalidade dos resultados já existentes de outras instâncias.

DISCRETE DISPERSION AND UNBALANCED LAGRANGEAN DECOMPOSITION FOR POINT-FEATURE CARTOGRAPHIC LABELING PROBLEM

ABSTRACT

This work concerns to the Point-Feature Cartographic Labeling Problem (PFCLP), which is a NP-Hard combinatorial problem. It is considered that when all points must be labeled and overlaps are inevitable, the map can be more readable if overlapping labels are placed in a dispersive way, i. e., overlapping labels are distant from each other. Thus, we present a Discrete Dispersion approach, generally used in the Facility Location Problem, which considers distance values between candidate positions. A constructive genetic algorithm to obtain results in more complex instances is likewise presented. Finally, we propose an unbalanced Lagrangean Decomposition, which achieved the optimal solution of several instances of PFCLP, and proved the optimality of various results of literature.

LISTA DE FIGURAS

Pág.

Figura 2.1 – Ilustração das três características cartográficas. 5
Figura 2.2 - Sobreposições de rótulos indicadas por setas. 6
Figura 2.3 - Padronização cartográfica proposta por Christensen et al. (1995). 7
Figura 2.4 - Grafo de conflitos para o PRCP: (a) Problema com 4 pontos e (b) seu grafo de conflitos correspondente. 8
Figura 2.5 – Comparação entre PMNRSC e PMNC. 11
Figura 3.1 – Distância entre posições candidatas são pesos de arestas no grafo de conflitos correspondente. 18
Figura 3.2 – Exemplo da abordagem dispersiva, quando não é possível evitar um conflito. 19
Figura 3.3 – Proporção de rótulos livres dos modelos testados. 27
Figura 4.1 – Exemplos de esquema (a) e estrutura (b). 39
Figura 4.2 – Pseudocódigo do AGC. 42
Figura 4.3 – Pseudocódigo da Busca Local 1. 44
Figura 4.4 – Pseudocódigo da Busca Local 2. 45
Figura 4.5 – Pseudocódigo da Busca Local 3. 46
Figura 4.6 – Pseudocódigo do operador de Mutação. 47
Figura 5.1 – Pseudocódigo do algoritmo Coalesce Vértices. 62
Figura 5.3 – Pseudocódigo do algoritmo Refina Particionamento. 64
Figura 5.4 – Exemplo de particionamento multinível com $k = 3$. 65

LISTA DE TABELAS

Pág.

Tabela 3.1 – Instâncias de 505 pontos..... 24
Tabela 3.2 – Instâncias de 5046 pontos..... 25
Tabela 3.3 – Resultados para 505 pontos..... 26
Tabela 3.4 – Resultados para 5046 pontos..... 26
Tabela 3.5 – Resultados da formulação do PRCP como *r*-Separação. 34
Tabela 4.1 – Número de conflitos em potencial das instâncias de testes..... 48
Tabela 4.2 – Parâmetros do AGC. 48
Tabela 4.3 – Resultados das instâncias de 505 pontos..... 50
Tabela 4.4 – Resultados das instâncias de 5046 pontos..... 51
Tabela 5.1 – Resultados utilizando o particionamento de Mauri et al. (2010)..... 66
Tabela 5.2 – Resultados utilizando o particionamento apresentado por Mauri et al. (2010). 68
Tabela 5.3 – Resultados com soluções ótimas obtidos com os particionamentos da Tabela 5.2. 69

LISTA DE SIGLAS E ABREVIATURAS

PRC	Problema de Rotulação Cartográfica
PRCP	Problema de Rotulação Cartográfica de Pontos
AG	Algortimo Genético
AGC	Algoritmo Genético Construtivo
DecLag	Decomposição Lagrangeana
SIG	Sistemas de Informação Geográfica
PMCIV	Problema do Máximo Conjunto Independente de Vértices
PMNRSC	Problema do Menor Número de Conflitos
PMNC	Problema do Menor Número de Conflitos
B&B	Branch & Bound
FMBPC	Formulação Matemática Baseada em Posições Candidatas
MD1	Modelo de Dispersão 1
MD2	Modelo de Dispersão 2
MD3	Modelo de Dispersão 3

LISTA DE SÍMBOLOS

G	Grafo representado pelo par (V, A)
V	Conjunto de vértices
A	Conjunto de arestas
N	Número de pontos de uma instância do PRCP
P_i	Quantidade de posições candidatas do ponto i
$S_{i,j}$	Conjunto de posições candidatas adjacentes a (i, j)
$d_{i,j,k,t}$	Distância entre (i, j) e (k, t)
D	Maior $d_{i,j,k,t}$ na instância
r	Raio
$L_u(r)$	conjunto de posições candidatas adjacentes a u, tal que $d_{u,v} < r$
d_{min}	Menor distância de conflitos em uma solução
$\delta(s_k)$	Ranking da solução s_k
s_{base}	Solução base
S_{guia}	Solução guia
P_α	População no instante α

SUMÁRIO

Pág.

1 INTRODUÇÃO 1
1.1 Motivação 1
1.2 Objetivos e contribuições 2
1.3 Organização da tese 3
2 PROBLEMA DE ROTULAÇÃO CARTOGRÁFICA 5
2.1 Representação do PRCP em grafos de conflitos 7
2.2 Abordagens do PRCP 10
2.3 Revisão bibliográfica para o PRCP 11
2.4 Considerações finais 15
3 MODELOS DE DISPERSÃO PARA O PRCP 17
3.1 Modelo de dispersão 1 (MD1) 20
3.2 Modelo de dispersão 2 (MD2) 22
3.2.1 Resultados para MD1, MD2 e FMBPC 23
3.3 Modelo de dispersão 3 (MD3) 28
3.4 Abordagem como *r*-Separação 29
3.4.1 Resultados do *r*-Separação para o PRCP 33
3.5 Considerações Finais 34
4 ALGORITMO GENÉTICO CONSTRUTIVO PARA DISPERSÃO DISCRETA NO PRCP 37
4.1 Estrutura do AGC para o PRCP 38
4.2 Operador de Recombinação 42
4.3 Operador de Mutação 43
4.4 Resultados 47
4.5 Considerações finais 52
5 DECOMPOSIÇÃO LAGRANGEANA DESBALANCEADA PARA O PRCP 53
5.1 Decomposição Lagrangeana para o PMNRSC 54
5.2 Particionamento desbalanceado 59

5.2.1 Algoritmo de particionamento desbalanceado 60
5.3 Resultados 65
5.4 Considerações finais 69
6 CONCLUSÃO 71
6.1 Trabalhos futuros 72
REFERÊNCIAS BIBLIOGRÁFICAS 75

1 INTRODUÇÃO

O posicionamento de rótulos é uma tarefa essencial no desenvolvimento de mapas, diagramas e objetos gráficos em geral. A atividade de rotular mapas automaticamente, em especial, é considerada um problema de grande complexidade, pois é necessário considerar diversos critérios quanto à escolha da localização, orientação, dimensões, característica a ser rotulada e etc. Este problema é denominado Problema de Rotulação Cartográfica (PRC) na literatura.

O PRC é um problema inerente à área de Sistemas de Informações Geográficas e geralmente é abordado de três formas diferentes: rotulação de características de linhas, rotulação de características de polígonos ou áreas e rotulação de características de pontos.

1.1 Motivação

Um fator comum às três abordagens de características geográficas do PRC é a necessidade de obedecer alguns critérios, definidos inicialmente em (IMHOF, 1975), para que uma dada rotulação seja considerada de boa qualidade. Estes critérios estão relacionados principalmente à visibilidade de rótulo e da característica geográfica a ser rotulada e alguns autores têm investigado a formalização do uso unificado destas métricas em seus modelos (EDMONSON et al., 1996; VAN DIJK et al., 2002;RYLOV; REIMER, 2014).

Este trabalho é focado no problema de rotular características de pontos, conhecido na literatura como Problema de Rotulação Cartográfica de Pontos (PRCP), cujo critério principal de avaliação da qualidade de soluções, tem sido o da proporção de rótulos livres de conflitos.

Como uma alternativa, neste trabalho é utilizado um novo critério de avaliação, que considera a distância entre os rótulos de uma solução. O objetivo do uso deste critério é obter soluções em que rótulos sejam posicionados de maneira esparsa, permitindo que em caso de não ser possível evitar conflitos, que a solução seja mais legível.

Para o emprego deste critério, é considerado o Problema de Dispersão Discreta, aplicado em geral ao Problema de Localização de Facilidades, em que o objetivo é dispersar facilidades “indesejadas”, ao selecioná-las para a localização.

Analogamente às facilidades, é indesejável que rótulos estejam posicionados de maneira muito próxima, com um pior caso de uma total sobreposição destes rótulos e considerando isto, é apresentada neste trabalho uma nova abordagem para o PRCP, que utiliza o conceito de dispersão discreta para os modelos propostos, e possui como principal critério de avaliação das soluções, a menor distância entre os rótulos posicionados.

Utilizando esta nova abordagem, são propostas formulações matemáticas de programação linear, para a resolução analítica do problema.

1.2 Objetivos e contribuições

Inicialmente é apresentada uma formalização do PRCP, de acordo com o problema de *p*-Dispersão apresentado por Moon e Chaudhry (1984), utilizando uma modelagem de programação matemática, com o objetivo de obter soluções esparsas de posicionamento de rótulos e consequentemente, evitar sobreposições. Para a análise das soluções destes modelos, é introduzido um novo critério de avaliação da qualidade das soluções do PRCP. Este novo critério é a medida de dispersão, que trata-se da máxima menor distância entre rótulos posicionados.

Em seguida é proposta, uma reformulação de restrições deste modelo, que permitiu melhorar os resultados de rótulos posicionados livres de conflito. Além disso, é apresentado um modelo matemático que considera tanto o objetivo de dispersão, quanto o objetivo de evitar conflitos.

Como uma investigação de métodos para obter soluções de instâncias em que *solvers* comerciais têm dificuldade de resolver, como instâncias com grande quantidade de pontos, é implementado um Algoritmo Genético Construtivo, que

mostrou um bom desempenho na literatura. Nesta implementação é utilizada a função objetivo do terceiro modelo de dispersão.

Para obter soluções através de métodos exatos, que possibilitem provar a otimalidade dos resultados, é apresentada a implementação da técnica de Decomposição Lagrangeana.

Na aplicação desta técnica, é proposta uma abordagem em que os subproblemas obtidos através da decomposição do problema original, possuem tamanho inomogêneo.

A nova abordagem foi comparada com a Decomposição Lagrangeana proposta por Mauri et al. (2010), que trata-se uma formulação matemática para o PRCP, decomposta homogeneamente no sentido Lagrangeano, que apresentou bons resultados na literatura.

Resultados de testes apresentados neste trabalho mostram que esta nova abordagem permite obter melhores resultados de soluções e de tempo de execução, quando comparados à versão balanceada.

1.3 Organização da tese

O restante deste trabalho de tese está organizado como se segue:

- O Capítulo 2 introduz o Problema de Rotulação Cartográfica de Pontos, suas principais abordagens, e apresenta uma revisão da literatura sobre o assunto;

- No Capítulo 3 é apresentada a abordagem de Dispersão Discreta para o Problema de Rotulação Cartográfica de Pontos, bem como os modelos propostos e resultados;

- No Capítulo 4 é discutido o Algoritmo Genético Construtivo implementado para um dos modelos de dispersão, e seus resultados são comparados;

- O Capítulo 5 apresenta a Decomposição Lagrangeana desbalanceada, para uma variação do Problema de Rotulação Cartográfica de Pontos. Os resultados desta abordagem são comparados com a versão balanceada.

- No Capítulo 6 são apresentadas as considerações finais e os trabalhos futuros.

2 PROBLEMA DE ROTULAÇÃO CARTOGRÁFICA

O Problema de Rotulação Cartográfica refere-se ao problema de rotular automaticamente mapas, diagramas, grafos ou qualquer objeto gráfico. É um problema comum da área de Sistemas de Informação Geográfica (SIG) e é abordado considerando três características cartográficas principais: linhas (representando rios, estradas, etc), polígonos (representando lagos, distritos, construções, etc) e pontos (cidades, montanhas, etc). Exemplos das três características cartográficas são mostrados na Figura 2.1.

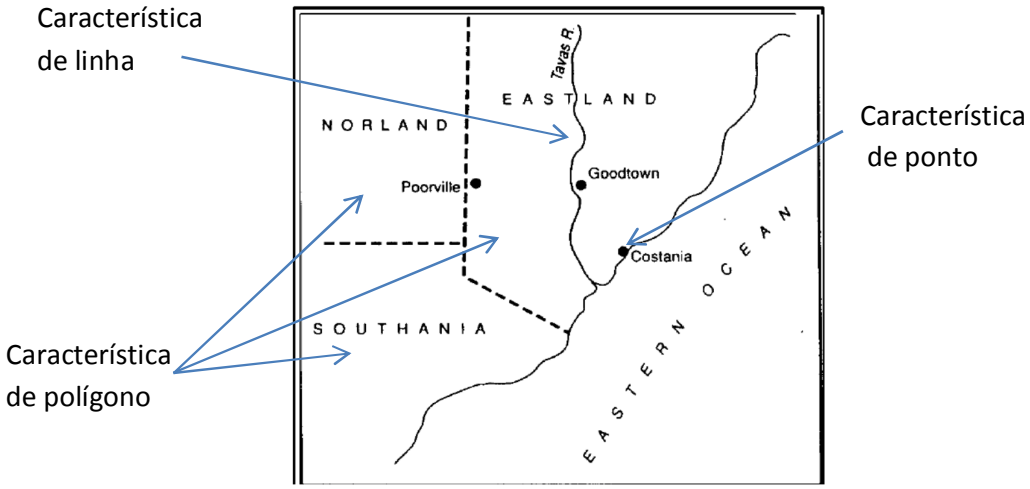

Figura 2.1 – Ilustração das três características cartográficas.

Assume-se que sobreposições de rótulos devem ser evitadas, tendo em vista a legibilidade do objeto gráfico a ser rotulado e dos rótulos posicionados. A Figura 2.2 apresenta um exemplo de sobreposição de rótulos no mapa do Estado do Espírito Santo.

Modelos de problemas de rotulação em geral são classificados em duas categorias principais: rótulos de posição fixa e rótulos deslizantes (VAN KREVELD et al. 1999). A primeira classificação refere-se a modelos em que o espaço onde rótulos podem ser posicionados é discretizado. Enquanto a segunda classificação está relacionada à versão contínua da modelagem do posicionamento de rótulos.

Figura 2.2 - Sobreposições de rótulos indicadas por setas.

Fonte: Ribeiro e Lorena (2006).

Este trabalho é focado no problema de rotular automaticamente características de pontos, conhecido na literatura como o Problema de Rotulação Cartográfica de Pontos (PRCP), considerando que as possíveis localizações de rótulos são fixas e conhecidas previamente.

Christensen et al. (1995) propuseram uma padronização cartográfica para discretizar a área de posicionamento de rótulos em torno dos pontos, em subdivisões conhecidas como posições candidatas. Este modelo utiliza um valor de prioridade para determinar as melhores opções, dentre as posições candidatas, para posicionar o rótulo. A Figura 2.3 ilustra esta padronização considerando quatro e oito posições candidatas respectivamente, utilizando valores de prioridade, com o menor número indicando a melhor posição para rotulação.

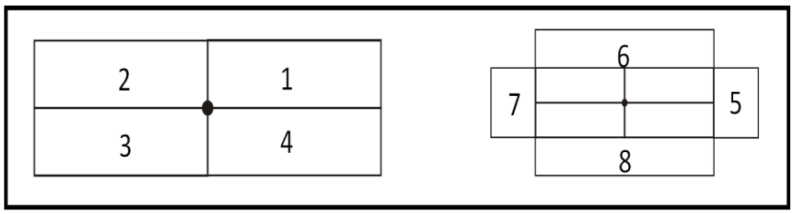

Figura 2.3 - Padronização cartográfica proposta por Christensen et al. (1995).

A Seção 2.1 discute a utilização desta padronização cartográfica de posições candidatas em um grafo de conflitos e a solução do Problema do Máximo Conjunto Independente de Vértices como uma solução para o PRCP.

2.1 Representação do PRCP em grafos de conflitos

Vários problemas que envolvem lógica matemática e computacional, como problemas de otimização combinatória, podem ser descritos através da modelagem de relações entre elementos de um mesmo conjunto. Uma estrutura na qual estes elementos são relacionados, é a de grafos.

Um grafo G é geralmente definido como $G = (V, A)$, com V um conjunto finito não vazio de elementos denominados vértices e A um conjunto de pares não ordenados de elementos distintos de V denominados arestas, definido por $A(V) = \{a(u, v) | u, v \in V, u \neq v\}$ (DIESTEL, 2000), de forma que o par $G = (V, A)$ com $A \subseteq A(V)$ é chamado um grafo em V.

Um tipo especial de grafos, cujo conceito é importante neste trabalho, é o grafo de conflitos, utilizado para representar relações lógicas entre variáveis binárias. Em problemas assim modelados, cada variável é mapeada a um vértice no grafo, e considera-se que vértices adjacentes estão em conflito, implicando que as variáveis que correspondem a estes vértices não podem assumir o mesmo valor.

O uso de variáveis binárias e a representação de seus conflitos são úteis para a modelagem do PRCP, pois a seleção de uma posição candidata da padronização cartográfica de Christensen et al. (1995), para a rotulação de um

ponto, pode ser representada por uma variável binária, em que se a posição candidata desta variável for selecionada, seu valor é 1, e caso contrário, é atribuído o valor 0.

Além da representação da seleção de posições candidatas, o grafo de conflitos é utilizado para modelar as possíveis sobreposições de rótulos, em que vértices adjacentes, e consequentemente em conflito, indicam sobreposições caso sejam selecionados simultaneamente para a rotulação. Um exemplo da representação de posições candidatas e suas possíveis sobreposições, em grafos de conflitos, é apresentado na Figura 2.4.

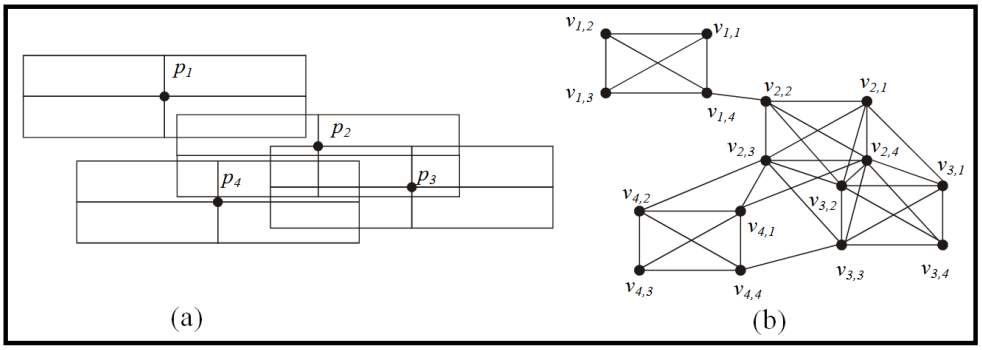

Figura 2.4 - Grafo de conflitos para o PRCP: (a) Problema com 4 pontos e (b) seu grafo de conflitos correspondente.

A Figura 2.4 (a) apresenta uma instância do PRCP com quatro pontos (p_i) e quatro posições candidatas para cada ponto, representadas pelos retângulos de acordo com a padronização cartográfica de Christensen et al. (1995). A Figura 2.4 (b) apresenta o grafo de conflitos correspondente, com o vértice $v_{i,j}$ a posição candidata $j \in \{1,\ldots,4\}$ do ponto $i \in \{1,\ldots,4\}$. Para cada sobreposição de posições candidatas em (a) existe uma aresta de conflito correspondente no grafo de (b). Note que, como cada ponto deve possuir apenas um rótulo, também existem arestas de conflitos entre posições candidatas de um mesmo ponto.

Outro problema modelado em grafo de conflitos muito conhecido é o Problema do Máximo Conjunto Independente de Vértices (PMCIV). Dado um grafo $G = (V, A)$, um conjunto $S \subseteq V$ é independente se para qualquer par de vértices $(v, s) \in S$, $\nexists\, a(v, s) \in A$, isto é, não existem arestas entre os vértices pertencentes ao conjunto *S.* O PMCIV é o problema de encontrar o conjunto *S* de maior cardinalidade.

Zoraster (1990) propôs a resolução do PRCP como um PMCIV, considerando que obter o conjunto *S* de maior cardinalidade, é equivalente a obter a maior quantidade possível de pontos rotulados sem conflitos, em um modelo do PRCP em grafo de conflitos.

O modelo de programação inteira do PRCP como PMCIV é descrito abaixo.

$$Max \sum_{x_{i,j} \in V} x_{i,j} \tag{2.1}$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} \leq 1 \qquad \forall i = \{1,...,N\}. \tag{2.2}$$

$$x_{i,j} + x_{k,t} \leq 1 \qquad \forall i = \{1,...,N\} \quad \forall j = \{1,...,P_i\} \quad (k,t) \in S_{i,j} \tag{2.3}$$

$$x_{i,j}, x_{k,t} \in \{0,1\}, \forall x_{i,j}, x_{k,t} \in V \tag{2.4}$$

A função objetivo visa maximizar a quantidade de posições candidatas selecionadas para rotulação. A variável de decisão $x_{i,j}$ é utilizada para indicar a seleção ou não de uma posição candidata *j* do ponto *i* – deste ponto em diante representada pelo par (i, j) – $\forall\ i = \{1,...,N\}$ e $\forall\ j = \{1,...,P_i\}$, onde *N* é o número de pontos e P_i é a quantidade de posições candidatas do ponto *i*.

A primeira restrição do problema assegura que apenas uma posição candidata (i, j) deve ser selecionada. Observa-se que neste caso, alguns pontos podem não ser rotulados.

A restrição (2.3) considera todas as arestas do grafo de conflitos, indicando que para cada par de posições candidatas adjacentes, no máximo uma delas deve ser selecionada, onde $S_{i,j}$ é o conjunto de posições candidatas adjacentes a (i, j). A restrição (2.4) indica os limites das variáveis inteiras.

A abordagem como PMCIV é a primeira a utilizar a representação em grafos de conflitos. Porém, devido o aumento do interesse na área de PRC em geral, novas abordagens foram desenvolvidas. A Seção 2.2 apresenta estas abordagens.

2.2 Abordagens do PRCP

Existem três principais abordagens para o PRCP na literatura: Problema do Máximo Conjunto Independente de Vértices (PMCIV), Problema do Máximo Número de Rótulos Sem Conflitos (PMNRSC) e Problema do Menor Número de Conflitos (PMNC).

Como discutido na seção anterior, na abordagem como PMCIV, o objetivo é obter o maior conjunto de pontos rotulados, de maneira que não existam conflitos entre eles, e como a maioria dos problemas práticos são muito complexos, alguns pontos podem não ser rotulados.

No PMNRSC, todos os pontos devem ser rotulados maximizando o número de rótulos livres de conflitos, porém a legibilidade do objeto cartográfico não é observada nesta abordagem.

O PMNC foi proposto como uma alternativa que considera a legibilidade, ao minimizar a quantidade de conflitos na solução, no processo de rotular todos os pontos. A Figura ilustra a diferença entre PMNRSC e PMNC.

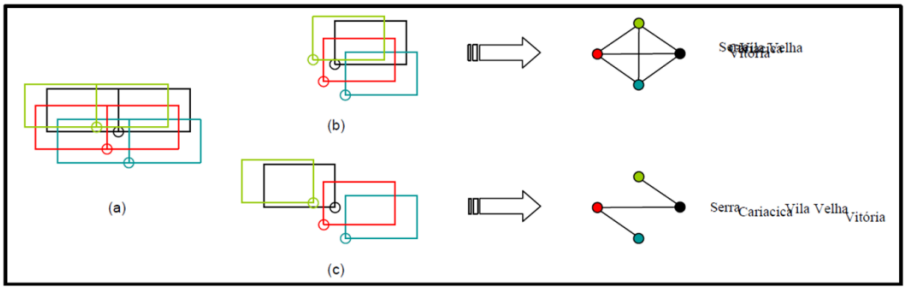

Figura 2.5 – Comparação entre PMNRSC e PMNC.
Fonte: Ribeiro (2007).

As Figura 2.5 (b) e Figura 2.5 (c) apresentam duas possíveis soluções para a instância da Figura 2.5 (a). Nestas duas soluções, todos os pontos estão rotulados em conflitos e por isto, são equivalentes do ponto de vista do PMNRSC. No entanto, se considerar como PMNC, a solução da Figura 2.5 (b) é pior que a solução da Figura 2.5 (c), pois apresenta um maior número de arestas de conflitos e consequentemente neste caso, os rótulos são menos legíveis.

Técnicas de solução para o PMNRSC e o PMNC tendem a atuar diretamente na estrutura do problema, isto é, considerando apenas as posições candidatas e seus possíveis conflitos, ao rotular pontos. Estre trabalho apresenta uma abordagem de dispersão discreta, já utilizada na literatura para o Problema de Localização de Facilidades, visando dispersar os rótulos posicionados para obter uma melhor legibilidade e evitar conflitos consequentemente.

Antes de discutir a abordagem proposta, é apresentada na Seção 2.3 uma revisão bibliográfica sobre o PRCP, e o modelo de PMNC utilizado para comparação com a nova abordagem apresentada no Capítulo 3.

2.3 Revisão bibliográfica para o PRCP

Existem diversos trabalhos na literatura atinentes às três principais abordagens do PRCP. Com relação ao PMCIV, Zoraster (1990) apresentou uma formulação

de programação linear inteira binária e propôs restrições com posições candidatas fictícias que penalizam com alto custo a função objetiva.

Strijk et al. (2000) propuseram formulações matemáticas utilizando restrições de corte com inequações para todas as cliques máximas no grafo de conflitos. Para instâncias grandes, aplicaram várias heurísticas: *Simulated Annealing*, *Diversified Neighborhood Search*, *k*-opt, e Busca Tabu.

Agarwal et al. (1998) propuseram algoritmos aproximativos para rotular sem conflitos o máximo número de pontos, computando o maior subconjunto de retângulos que não se intersectam, utilizando dimensões variadas de largura.

Verweij e Aardal (1999) apresentaram um algoritmo *branch-and-cut* e resultados para instancias com 950 pontos (3800 vértices no grafo de conflitos). Os autores sugerem que é possível melhorar o algoritmo utilizando uma técnica recursiva. Também é apresentada uma variação desta técnica que é baseada em decomposição de caminhos.

Ribeiro et al. (2011) apresentaram uma decomposição Lagrangeana para o problema e apresentaram soluções ótimas para quase todas as instâncias de grande escala da literatura. Os autores sugerem que o método pode ser uma boa alternativa para resolver outros problemas baseados em PMCIV.

Com relação ao PMNRSC, Christensen et al. (1995) propuseram dois algoritmos baseados em uma forma de decida de gradiente e Simulated Annealing.

Yamamoto et al. (2002) propuseram um algoritmo Busca Tabu que obteve bons resultados em bancos de dados reais.

Um algoritmo exato e um Algoritmo Genético Construtivo foram desenvolvidos por Yamamoto e Lorena (2005). O algoritmo exato não obteve bons resultados em instâncias com mais de 25 pontos, enquanto o algoritmo genético obteve bons resultados em instâncias com até 1000 pontos.

Alvim e Taillard (2005) utilizaram as instâncias propostas por Yamamoto et al. (2002) para testar a técnica POPMUSIC em instâncias grandes. A POPMUSIC inicialmente constrói subproblemas a partir do problema principal. Estes problemas são resolvidos e então agregados em uma solução para o problema principal. O problema principal é alterado e o processo é reiniciado até que um dado critério de parada seja alcançado.

Mauri et al. (2010) propuseram o primeiro modelo matemático para o PMNRSC e uma decomposição Lagrangeana para resolver o problema em instâncias de até 1000 pontos. Os resultados computacionais apresentaram soluções ótimas para diversas instâncias da literatura.

O PMNC foi introduzido por Ribeiro e Lorena (2006) como uma nova abordagem para melhorar a legibilidade quando todos os pontos devem ser rotulados. Os autores apresentaram uma formulação matemática e algumas heurísticas de relaxação Lagrangeana.

Cravo et al. (2008) propuseram um Procedimento de Busca Adaptativa Aleatória Gulosa (GRASP) e obtiveram melhores resultados que outras técnicas disponíveis na literatura.

Ribeiro e Lorena (2008) apresentaram duas novas formulações para o PMNC e uma relaxação Lagrangeana com clusters, que obtiveram melhores resultados que outras técnicas da literatura. A principal diferença entre as formulações é como o grafo de conflitos é construído: uma formulação é baseada em posições candidatas enquanto a outra é baseada em posições candidatas e pontos.

Duas formulações de dispersão discreta para o PRCP, descritas neste trabalho de tese, foram apresentadas por Gomes et al. (2013). Estas formulações utilizam distâncias entre posições candidatas nos grafos de conflitos, e obtiveram bons resultados, tanto de dispersão quanto de proporção de rótulos livres de conflitos.

Gomes et al. (2014) propuseram um novo modelo, derivado da abordagem de dispersão discreta, que considera um parâmetro de distância mínima entre rótulos posicionados *r*. Esta abordagem, já utilizada no Problema de Localização de Facilidades, é proposta como uma adaptação do Problema de *p*-Dispersão para o PRCP, e resultados do modelo são também apresentados.

Neste trabalho, é utilizada a Formulação Matemática Baseada em Posições Candidatas (FMBPC) proposta por Ribeiro e Lorena (2008) para comparação dos modelos apresentados. A FMBPC é formulada por

$$Min \sum_{i=1}^{N} \sum_{j=1}^{P_i} \left(w_{i,j} x_{i,j} + \sum_{(k,t) \in S_{i,j}} y_{i,j,k,t} \right) \tag{2.5}$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \qquad \forall i = \{1,...,N\} \tag{2.6}$$

$$x_{i,j} + x_{k,t} - y_{i,j,k,t} \leq 1 \qquad \forall i = \{1,...,N\} \quad \forall j = \{1,...,P_i\} \quad (k,t) \in S_{i,j} \tag{2.7}$$

$$x_{i,j}, x_{k,t}, y_{i,j,k,t} \in \{0,1\} \qquad \forall i = \{1,...,N\} \quad \forall j = \{1,...,P_i\} \quad (k,t) \in S_{i,j}. \tag{2.8}$$

Na formulação (2.5) – (2.8), $w_{i,j}$ é o peso que indica a prioridade cartográfica de Cristensen et al. (1995). A função objetivo (2.5) tenta minimizar a quantidade

de conflitos, indicados pelas variáveis $y_{i,j,k,t}$, com valor 1 caso exista um conflito entre as posições candidatas (i, j) e (k, t). As restrições (2.6) asseguram que uma posição candidata seja selecionada para cada ponto. As restrições (2.7) atribui o valor da variável $y_{i,j,k,t}$, no caso de haver conflito entre as posições candidatas (i, j) e (k, t).

Considerando que instâncias do PRCP podem gerar grafos de conflitos com um grande número de arestas e algumas informações redundantes, Wagner et al. (2001) apresentaram um método para a redução de problemas de rotulação sem alterar o conjunto de soluções ótimas. Este método consiste das seguintes três regras:

- Se o ponto $i = \{1,...,N\}$ possui uma posição candidata $j = \{1,...,P_i\}$, sem nenhum conflito em potencial, adicione j como parte da solução e elimine as outras posições candidatas de i;

- Se o ponto $i = \{1,...,N\}$ possui uma posição candidata $j_1 = \{1,...,P_i\}$, que possui apenas um conflito com outra posição candidata $t_1 = \{1,...,P_k\}$; e o ponto k possui uma posição candidata $t_2 = \{1,...,P_k\}: t_2 \neq t_1$, que está sobreposto somente por uma posição candidata $j_2 = \{1,...,P_i\}: j_2 \neq j_1$, adicione j_1 e t_2 como parte da solução e elimine todas as outras posições candidatas de i e k.

- Se o ponto $i = \{1,...,N\}$ possui apenas uma posição candidata $j = \{1,...,P_i\}$, e as sobreposições de j constituem uma clique no grafo de conflitos, adicione j como parte da solução e elimine todas as outras posições da clique.

Estas regras devem ser reaplicadas sempre que uma nova posição candidata for adicionada à solução.

2.4 Considerações finais

Neste capítulo apresentou-se uma introdução do Problema de Rotulação Cartográfica de Pontos, bem como suas principais abordagens como um

problema de otimização combinatória, que utilizam em geral, grafos de conflitos como estrutura de dados principal.

Além disso, foi apresentada uma revisão da bibliografia sobre estas abordagens, utilizada para a construção das contribuições deste trabalho de tese.

Nestas abordagens do PRCP, evitar conflitos é um objetivo que é considerado diretamente na modelagem do problema. No Capítulo 2 a seguir, é introduzida a abordagem de dispersão discreta para o PRCP, proposta neste trabalho, que utiliza a distância entre rótulos com sobreposições em potencial, no grafo de conflito e tenta maximizar a menor distância entre rótulos, evitando sobreposições indiretamente. Baseada nesta abordagem, modelos de programação matemática são propostos e resultados computacionais são apresentados.

3 MODELOS DE DISPERSÃO PARA O PRCP

Este capítulo introduz a nova abordagem proposta, que utiliza o conceito de Dispersão Discreta, e apresenta as novas formulações para o PRCP. A abordagem de problemas de Otimização Combinatória como Dispersão Discreta surgiu de uma especialização do Problema de Localização de Facilidades.

Exemplos de problemas de dispersão de facilidades incluem dispersão de centros de reabilitação criminal, dispersão de usinas nucleares, dispersão de lojas de uma mesma franquia, etc (CURTIN; CHURCH, 2006).

Erkut e Neuman (1990) apresentaram uma classificação para problemas de dispersão em geral, baseada no tipo de métrica de dispersão utilizada para a modelagem do problema. As classes são:

- MaxMinMin: Problemas que se enquadram nesta classe são denominados na literatura de *p*-Dispersão (MOON; CHAUDRHRY, 1984), que é o problema de localizar *p* facilidades, de maneira que a distância mínima entre cada par de facilidades adjacentes seja maximizada. Uma variação deste modelo dispersivo, que surgiu do *p*-Dispersão é o problema de *r*-Separação (ERKUT et al., 1996), nesta abordagem a distância mínima entre as facilidades é dada como um parâmetro *r*.

- MaxSumMin: Este problema é o de maximizar conjuntos de distâncias mínimas entre facilidades, ao invés de somente a menor distância entre facilidades. Este problema é conhecido na literatura como o problema de *p*-Defesa (MOON; CHAUDRHRY, 1984).

- MaxMinSum: Problemas com esta classificação também têm como objetivo obter a máxima soma das menores distâncias entre facilidades. Porém, neste caso, devem ser consideradas as distâncias entre todas as facilidades, e não somente entre facilidades adjacentes (ERKUT; NEUMAN, 1990).

- MaxSumSum: É uma abordagem mais global do problema MaxMinSum, pois não se quer obter a máxima soma das distâncias mínimas, mas sim a máxima soma de arestas de distância em geral. Em geral, quando se utiliza esta abordagem, tem-se o objetivo de maximizar a distância entre *p* facilidades a serem localizadas e um dado conjunto de facilidades fixas, já pré-localizadas na instância (KUBY, 1987).

A abordagem dispersiva apresentada neste trabalho considera o tipo MaxMinMin, ou *p*-Dispersão, para a modelagem do PRCP, e como sugerido por Moon e Chaudhry (1984), utiliza-se a menor distância obtida nos resultados da resolução do problema, como métrica de avaliação das soluções.

No contexto do PRCP, é desejável que os rótulos sejam posicionados esparsamente, de maneira que conflitos sejam evitados se possível. Com este propósito, é considerada a distância entre as posições candidatas de pontos diferentes, no grafo de conflitos, conforme ilustrado na Figura 3.1 – Distância entre posições candidatas são pesos de arestas no grafo de conflitos correspondente.

.

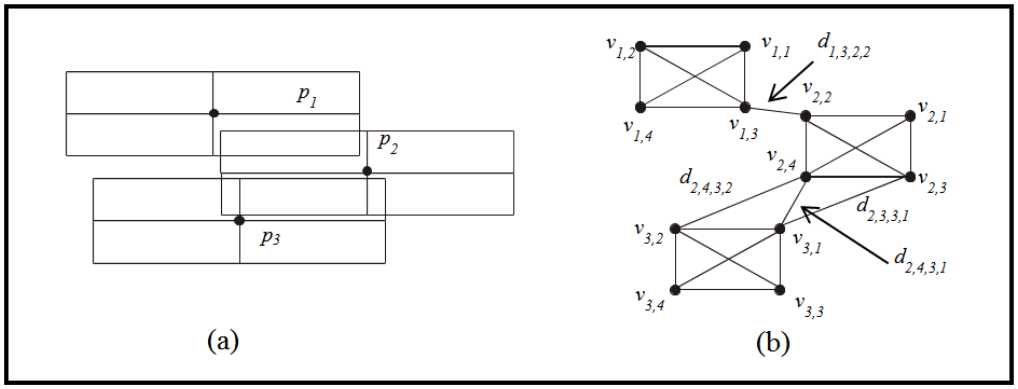

Figura 3.1 – Distância entre posições candidatas são pesos de arestas no grafo de conflitos correspondente.

A Figura 3.1 apresenta um conjunto de três pontos, com quatro posições candidatas para cada ponto (a), e o grafo de conflitos correspondente, com o vértice $v_{i,j}$ representando a posição candidata *j* do ponto *i* (b). Os pesos das arestas são as distâncias $d_{i,j,k,t}$ entre posições candidatas $v_{i,j}$ e $v_{k,t}$. O índice *j* representa o *rank* da posição candidata de acordo com a padronização de Christensen et al. (1995).

Quando se quer rotular características de pontos e não é possível evitar sobreposições, escolher rótulos que produzam uma maior distância de conflito, pode permitir que a solução seja mais legível. A Figura 3.2 ilustra um processo de seleção de rótulos, quando não é possível evitar uma sobreposição.

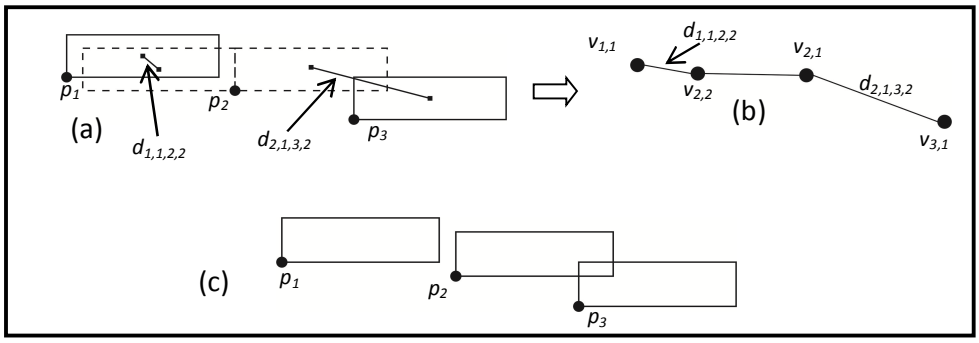

Figura 3.2 – Exemplo da abordagem dispersiva, quando não é possível evitar um conflito.

A Figura 3.2 apresenta uma instância com três pontos, dos quais, p1 e p3 já são pontos rotulados (a). Todos os conflitos são mapeados para o grafo de conflitos em (b). Observa-se que as distâncias são utilizadas apenas em arestas de conflitos entre posições candidatas de pontos distintos e como todas as posições candidatas de p2 possuem conflitos, as duas opções para rotular produzem o mesmo resultado para a função objetivo da formulação (2.5)–(2.8). Por outro lado, se for levado em consideração, que a posição candidata que está em conflito com um valor de distância maior, possui uma menor área de rótulo sobreposta, do ponto de vista da legibilidade, a posição candidata v3,1 é a melhor escolha (c).

Como exemplificado na Figura 3.2, o valor de distância entre rótulos pode ser utilizando como uma medida para avaliar uma solução do ponto de vista da dispersão, e por isto, utiliza-se como medida de dispersão, a menor distância obtida, como critério de comparação entre soluções.

Na Seção 3.1 é formalizada a abordagem do PRCP como um problema de *p*-Dispersão, denominado aqui de Modelo de Dispersão 1 (MD1). Após esta consideração, é apresentada na Seção 3.2 uma formulação desenvolvida a partir do MD1, que obteve melhores resultados de proporção de rótulos livres de conflitos, denominada aqui de Modelo de Dispersão 2 (MD2). Nessa seção os resultados de MD1 e MD2 são apresentados e comparados aos resultados da FMBPC (formulação (2.5) – (2.8)), apresentada no Capítulo 2.

Na Seção 3.3, apresenta-se um modelo que leva em conta tanto o objetivo de dispersão de rótulos quanto o objetivo de minimizar o número de conflitos. A função objetivo deste modelo é utilizada no Algoritmo Genético Construtivo (AGC) proposto neste trabalho e, portanto, seus resultados são apresentados e comparados no Capítulo 4, onde o AGC é discutido.

Outra abordagem que utiliza distâncias entre posições candidatas, que está relacionada diretamente ao *p*-Dispersão é a do problema de *r*-Separação. Esta relação é melhor discutida na Seção 3.4 e uma formulação desta abordagem para o PRCP e resultados são apresentados.

3.1 Modelo de dispersão 1 (MD1)

Considerando o PRCP como um problema estruturado em um grafo de conflitos com pesos em suas arestas, como apresentado na Figura 3.1, para cada posição candidata, é necessário decidir se esta será utilizada para posicionar um rótulo ou não. Esta decisão também será representada aqui pelo valor de uma variável binária $x_{i,j,} \ \forall \ i = \{1,\ldots,N\}$ e $j = \{1,\ldots,P_i\}$. Se a posição (i, j) é escolhida, tem-se que $x_{i,j} = 1$, e $x_{i,j} = 0$ caso contrário.

Para este modelo, todos os pontos devem ser rotulados e somente um rótulo deve ser posicionado para cada ponto. Esta restrição é modelada pela inequação

$$\sum_{j=1}^{P_i} x_{i,j}, \forall\, i = \{1,...,N\} \tag{3.1}$$

Também é necessário considerar as sobreposições de rótulos. Se não é possível evitar conflitos, é desejável que os rótulos sejam posicionados de maneira mais esparsa. A restrição a seguir utiliza os valores de distância $d_{i,j,k,t}$ entre posições candidatas (i, j) e (k, t):

$$\begin{aligned} & z - M(2 - x_{i,j} - x_{k,t}) \le d_{i,j,k,t} \qquad \forall i = \{1,...,N\} \\ & \qquad\qquad\qquad\qquad\qquad\qquad\quad \forall j = \{1,...,P_i\} \\ & \qquad\qquad\qquad\qquad\qquad\qquad\quad (k,t) \in S_{i,j}. \end{aligned} \tag{3.2}$$

$S_{i,j}$ é o conjunto de posições candidatas adjacentes a (i, j), M é um valor fixo tal que $M > d_{i,j,k,t}$, $\forall\ i = \{1,...,N\}$, $j = \{1,...,P_i\}$ e (k, t) $\in S_{i,j}$. O valor M é utilizado como uma recompensa, caso as posições candidatas referentes a esta restrição não serem selecionadas, permitindo z aumentar até $2M$. Se somente uma das posições candidatas for selecionada, z pode aumentar até M, que é maior que o valor de $d_{i,j,k,t}$, obtido caso as posições candidatas sejam selecionadas simultaneamente, gerando um conflito.

O MD1 é dado pela formulação de programação linear inteira mista apresentada a seguir:

$$Max \quad z \tag{3.3}$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \qquad \forall i = \{1,...,N\} \tag{3.4}$$

$$z - M(2 - x_{i,j} - x_{k,t}) \le d_{i,j,k,t} \qquad \forall i = \{1,...,N\} \tag{3.5}$$

$$\forall j = \{1,\ldots,P_i\}$$

$$(k,t) \in S_{i,j}$$

$$x_{i,j}, x_{k,t} \in \{0,1\} \qquad \forall i = \{1,\ldots,N\}$$

$$\forall j = \{1,\ldots,P_i\} \tag{3.6}$$

$$(k,t) \in S_{i,j}$$

$$z \geq 0. \tag{3.7}$$

A restrição (3.5) limita o valor de z em $2M + min(d_{i,j,k,t})$, onde $min(d_{i,j,k,t})$ é a menor distância entre rótulos em potencial conflito, para uma dada instância do PRCP. Como o objetivo é maximizar z, é preferível priorizar conflitos com maiores valores de $d_{i,j,k,t}$, quando não é possível evitar sobreposições. O conjunto de restrições (3.6) define os limites das variáveis inteiras da formulação.

3.2 Modelo de dispersão 2 (MD2)

Para o segundo modelo de dispersão, as distâncias entre rótulos em conflito são utilizadas, como uma recompensa para o posicionamento de rótulos mais dispersos. Da mesma forma que no DM1, a função objetivo maximiza a distância mínima z entre os rótulos posicionados, e restrições na forma das equações (3.4) são utilizadas para assegurar que cada ponto tenha uma posição candidata selecionada para rotulação. Para evitar sobreposições, são consideradas restrições com pesos, conforme dado pela equação (3.5). No entanto, ao invés do valor fixo M, foram utilizados os valores de distância $d_{i,j,k,t}$ como recompensa por prevenir um conflito entre (i, j) e (k, t), de forma que as restrições são reformuladas como se segue:

$$z \leq d_{i,j,k,t}(2 - x_{i,j} - x_{k,t}) \qquad \forall i = \{1,\ldots,N\}$$

$$\forall j = \{1,\ldots,P_i\} \tag{3.8}$$

$$(k,t) \in S_{i,j}.$$

Ao maximizar o valor de z, é mais vantajoso atribuir 0 às variáveis $x_{i,j}$ e $x_{k,t}$ de posições candidatas em possível conflito, de maneira que z possa aumentar até $2d_{i,j,k,t}$, evitando conflitos. O problema é formulado a seguir.

$$Max \quad z \tag{3.9}$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \qquad \forall i = \{1,\ldots,N\} \tag{3.10}$$

$$z \leq d_{i,j,k,t}(2 - x_{i,j} - x_{k,t}) \qquad \forall i = \{1,\ldots,N\}, \quad \forall j = \{1,\ldots,P_i\}, \quad (k,t) \in S_{i,j} \tag{3.11}$$

$$x_{i,j}, x_{k,t} \in \{0,1\} \qquad \forall i = \{1,\ldots,N\}, \quad \forall j = \{1,\ldots,P_i\}, \quad (k,t) \in S_{i,j} \tag{3.12}$$

$$z \geq 0. \tag{3.13}$$

Restrições (3.11) limitam o valor de z ao máximo de $2min(d_{i,j,k,t})$, obtido quando se tem uma solução livre de conflitos. Restrições (3.12) e (3.13) definem o domínio das variáveis de decisão.

3.2.1 Resultados para MD1, MD2 e FMBPC

Esta seção apresenta os resultados dos testes computacionais para os modelos MD1, MD2 e FMBPC, utilizando o *solver* CPLEX 12.1 (IBM, 2009) em conjuntos de 505 e 5046 pontos, considerando 4 posições candidatas para cada ponto, de acordo com a padronização proposta por Christensen et al. (1995).

Os dados utilizados foram obtidos de um conjunto de pontos correspondendo a cidades no mapa da Suíça, disponível na URL http://mistic.heig-

vd.ch/taillard/problemes.dir/problemes.html, e 12 instâncias foram geradas de ambos os conjuntos, ao variar as dimensões de altura e largura dos rótulos.

Os resultados dos modelos de Dispersão Discreta para o PRCP são comparados com a FMBPC, que apresentou bons resultados na literatura (RIBEIRO; LORENA, 2008) com relação à proporção de rótulos livres de conflitos. Os testes computacionais foram executados em um computador Intel Core i5, com 4 GB de RAM.

As tabelas Tabela 3.1 e Tabela 3.2 apresentam os detalhes dos dados das instâncias com 505 e 5046 pontos respectivamente, após aplicar o método de redução de Wagner et al. (2001). Os títulos das colunas são os mesmos em ambas as tabelas. As duas primeiras colunas apresentam altura e largura dos rótulos utilizados nestas instâncias. A terceira coluna apresenta o número de conflitos após o uso da técnica de redução. A quarta coluna apresenta o número de pontos rotulados no método de redução. As colunas restantes apresentam o número de restrições e variáveis para os modelos testados.

Tabela 3.1 – Instâncias de 505 pontos.

A	L	Conflitos	Redução	Restrições	Variáveis	
				MD1, MD2 e FMBPC	MD1 e MD2	FMBPC
2	24	1115	316	1620	1073	2187
2	32	1667	263	2172	1232	2898
3	16	1702	267	2207	1220	2921
2	42	2151	232	2656	1325	3475
2	48	2371	219	2876	1364	3734
3	24	2955	176	3460	1493	4447
4	16	3224	175	3729	1496	4719
3	28	3439	160	3944	1541	4979
4	18	3756	149	4261	1574	5329
3	32	3995	137	4500	1610	5604
4	21	4490	122	4995	1655	6144
4	24	5035	110	5540	1691	6725

Tabela 3.2 – Instâncias de 5046 pontos.

A	L	Conflitos	Redução	Restrições	Variáveis	
				MD1, MD2 e FMBPC	MD1 e MD2	FMBPC
2	24	14003	3101	19049	10882	24884
2	32	19350	2644	24396	12253	31602
3	16	20590	2584	25636	12433	33022
2	42	25572	2253	30618	13426	38997
2	48	29151	2075	34197	13960	43110
3	24	33591	1833	38637	14686	48276
4	16	34445	1856	39491	14617	49061
3	28	39633	1593	44679	15406	55038
4	18	39644	1626	44690	15307	54950
3	32	45762	1368	50635	15799	61387
4	21	46796	1383	51842	16036	62831
4	24	53914	1180	58960	16645	70558

A técnica de redução pôde rotular com sucesso ao menos 21.7% dos pontos nas instâncias de 505 pontos, e ao menos 23.3% dos pontos nas instâncias de 5046 pontos. Este pré-processamento das instâncias permitiu reduzir significativamente o número de conflitos, e consequentemente o tempo de execução nos testes.

Os testes do MD1 foram executados com M = 100 para as instâncias de 505 pontos e M = 15 para instâncias de 5046 pontos. A opção best-estimate foi escolhida para a estratégia de seleção de nós, nos testes de MD2. As tabelas Tabela 3.3 e Tabela 3.4 apresentam resultados de proporção de rótulos livres de conflitos e de dispersão para todas as 24 instâncias geradas. Para avaliar os resultados de dispersão, é utilizada a menor distância de conflito na solução, apresentada na coluna d_{min}. Os melhores resultados são destacados em negrito.

Tabela 3.3 – Resultados para 505 pontos.

T. Execução (*seg*)			Rótulos livres (%)			d_{min}		
FMBPC	**MD1**	**MD2**	**FMBPC**	**MD1**	**MD2**	**FMBPC**	**MD1**	**MD2**
0.09	0.98	0.12	100.00	100.00	100.00	-	-	-
0.28	2.72	1.31	100.00	100.00	100.00	-	-	-
0.14	2.58	1.17	100.00	100.00	100.00	-	-	-
1.59	6.06	3.85	99.20	99.20	99.20	10.04	**33.01**	9.05
2.02	4.83	4.29	99.20	99.20	**99.60**	30.01	**39.01**	9.05
2.41	19.78	2.29	98.40	93.80	**99.60**	11.01	**18.11**	9.21
15.97	174.98	5.79	96.40	85.15	**98.40**	6.08	**11.4**	9.05
4.28	66.75	4.41	97.20	88.30	**99.20**	3.16	**19.1**	9.05
6.92	438.22	4.04	95.20	77.23	**98.40**	3.16	**12.16**	9.21
6.01	41.59	34.62	95.60	85.7	**98.00**	3.16	**22.1**	9.05
17.87	227.48	6.32	93.40	79.21	**99.20**	2.23	**14.03**	9.21
193.86	227.75	17.07	90.50	64.55	**97.60**	1.41	**14.31**	9.05

Tabela 3.4 – Resultados para 5046 pontos.

T. Execução (*seg*)			Rótulos livres (%)			d_{min}		
FMBPC	**MD1**	**MD2**	**FMBPC**	**MD1**	**MD2**	**FMBPC**	**MD1**	**MD2**
9.23	10.37	10.37	99.64	93.20	**99.96**	1.41	**17.02**	9.05
27.30	20.59	20.59	99.24	90.90	**99.88**	2.23	**17.02**	9.05
52.82	32.09	32.09	98.31	87.80	**99.38**	1.41	**10.04**	9.05
32.71	28.21	28.21	98.65	83.70	**99.84**	2.23	**17.02**	9.05
68.08	58.32	58.32	98.07	77.50	**99.84**	1.41	**17.02**	9.05
5500	5500	5500	95.65	70.10	**99.6**	1.41	**13.03**	9.05
5500	5500	5500	96.11	75.40	**98.73**	1.41	6.32	**9.05**
5500	5500	5500	94.26	75.20	**99.44**	1.41	**13.15**	9.05
5500	5500	5500	92.35	67.70	**99.16**	1.41	6.32	**9.05**
5500	5500	5500	91.89	57.40	**99.58**	1.41	**12.16**	9.05
5500	5500	5500	89.73	49.90	**99.08**	1.41	6.32	9.05
5500	5500	5500	87.08	43.20	**98.85**	1.41	5.09	9.05

Como o MD1 é focado no problema de dispersão, o modelo obteve melhores resultados com relação à medida de desempenho utilizada. No entanto,

conforme a complexidade das instâncias aumenta apenas o FMBPC e MD2 obtiveram bons resultados para proporção de rótulos livres de conflitos, com o último apresentando melhores resultados. Traços nos resultados para d_{min}, a máxima menor distância obtida, implicam que todos os rótulos nas soluções foram posicionados livres de conflitos.

O tempo de execução dos testes foi limitado em 5500 segundos por causa da sobrecarga de memória na execução do CPLEX para as instâncias maiores. A Figura 3.3 sumariza os resultados de proporções de rótulos livres apresentados na Tabela 3.4. Observa-se que MD2, apresentou uma performance mais estável, conforme o número de possíveis conflitos aumenta, obtendo melhores resultados para todas as instâncias de 5046 pontos.

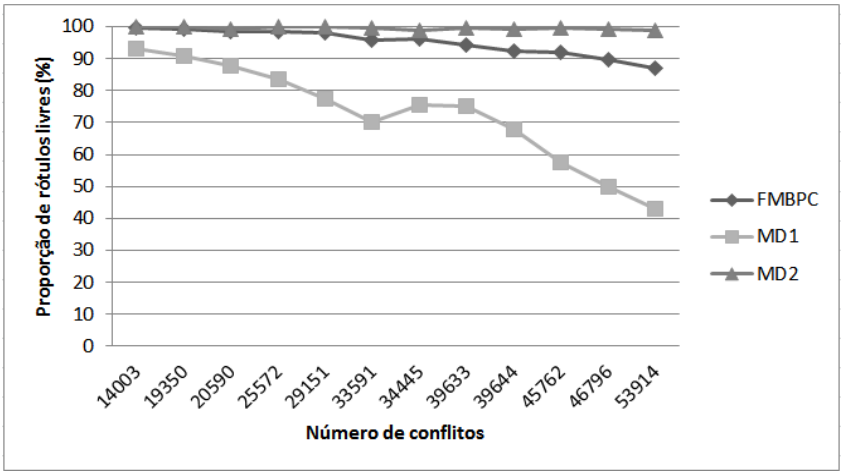

Figura 3.3 – Proporção de rótulos livres dos modelos testados.

Estes resultados são possíveis para MD2 com o uso da opção *best-estimate* do CPLEX, em que a estratégia para a escolha do próximo nó da árvore do Branch e Bound (B&B), a ser explorado, é a de selecionar o nó que produz o melhor valor estimado para a função objetivo, ao invés de escolher o nó que obtém o melhor limitante.

Considerando a inequação (3.11), o valor z de uma solução de programação linear para esta formulação é no máximo $2 \cdot min(d_{i,j,k,t})$, onde $min(d_{i,j,k,t})$ é a distância mínima entre dois rótulos em possível conflito em uma instância do PRCP. Em instâncias mais complexas, com maior número de conflitos, pode

não ser possível obter soluções sem conflitos. Em poucas iterações do B&B, esse valor decresce de $2 \cdot min(d_{i,j,k,t})$ para 0, e como o objetivo em MD2 é obter a máxima menor distância z, é possível que existam várias configurações de rótulos para uma instância com $z = 0$.

Considerando o número de conflitos, é mais interessante para o MD2 o uso do parâmetro *best-estimate* do CPLEX, por causa de sua característica de busca em profundidade. Este método de seleção utiliza um modelo estocástico para estimar quão bom é o arredondamento de variáveis específicas na árvore do B&B, para a solução final.

Como o método considera o efeito possível de mudanças na variável para a solução final, e não a solução em si, o algoritmo não para no melhor limitante $z = 0$, mas continua buscando bons resultados de configuração de rótulos.

3.3 Modelo de dispersão 3 (MD3)

Os modelos de dispersão apresentados até agora têm como objetivo obter a máxima menor distância entre as posições candidatas selecionadas para a rotulação, e ao dispersar rótulos, consequentemente evitar conflitos. Nesta seção, apresenta-se um modelo que leva em conta diretamente em sua função objetivo as abordagens de dispersão e de menor número de conflitos.

Para esta nova formulação, considera-se que ao tentar rotular, conflitos sejam evitados primariamente, e em caso de houver a necessidade de sobreposições, que sejam escolhidas as maiores distâncias de conflitos.

Uma das restrições deste problema é a de que todos os pontos devem ser rotulados, e por isso utilizam-se as equações da forma (3.10). Para indicar a existência de conflitos, utilizam-se as inequações (2.7), como um segundo conjunto de restrições. A formulação de MD3 é apresentada como se segue.

$$Max \sum_{(i,j)\in V(k,t)\in S_{i,j}} \sum \left(\frac{d_{i,j,k,t}}{D} y_{i,j,k,t} \right) + \sum_{(i,j)\in V(k,t)\in S_{i,j}} \sum (1 - y_{i,j,k,t}) \quad (3.14)$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \qquad \forall i = \{1,...,N\} \quad (3.15)$$

$$x_{i,j} + x_{k,t} - y_{i,j,k,t} \leq 1 \qquad \forall i = \{1,...,N\} \quad (3.16)$$

$$\forall j = \{1,...,P_i\}$$

$$(k,t) \in S_{i,j}$$

$$x_{i,j}, x_{k,t}, y_{i,j,k,t} \in \{0,1\} \qquad \forall i = 1...N \quad (3.17)$$

$$\forall j = 1...P_i$$

$$(k,t) \in S_{i,j}$$

A função objetivo tenta minimizar o número de conflitos com $y_{i,j,k,t} = 0$, aumentando o valor do segundo somatório duplo. Quando não é possível evitar sobreposições, conflitos com maiores valores de distância são priorizados no primeiro somatório duplo, onde D é o maior $d_{i,j,k,t}$ na instância. Resultados de testes desta formulação são apresentados em comparação com os resultados do Algoritmo Genético Construtivo proposto no Capitulo 4.

3.4 Abordagem como *r*-Separação

O *r*-Separação é descrito como o problema de encontrar em um conjunto de pontos, o maior subconjunto possível de pontos que estão separados por uma distância maior ou igual a um dado valor de distância *r*. Este problema tem aplicações em problemas de dispersão, quando se tem a priori a distância mínima entre os objetos a serem localizados, e é considerado NP-Completo, por causa de sua equivalência com o PMCIV.

Erkut et al. (1996) apresentaram uma formulação para o *r*-Separação como um problema de programação inteira, que utiliza a variável binária x_u representando a seleção do ponto candidato *u* para a localização de uma facilidade. Desta forma, $x_u = 1$ caso o ponto u seja selecionado para localizar a facilidade e $x_u = 0$ caso contrário.

Uma restrição para o problema de *r*-Separação é a de que, ao selecionar o ponto *u* pertencente ao conjunto *U* de pontos candidatos, é necessário assegurar sua separação dos demais em ao menos *r* unidades de distância.

Considerando $L_u(r)$ um conjunto de posições candidatas adjacentes a *u*, tal que $d_{u,v} < r$, $\forall\ v \in L_u(r)$, em que $d_{u,v}$ é a distância euclidiana entre as posições candidatas *u* e *v*, tem-se o conjunto de restrições

$$x_u + x_v \leq 1, \qquad \forall u = \{1,...,U\} \qquad \forall v \in L_u(r). \tag{3.18}$$

A inequação (3.1) assegura que a posição *u* e suas posições adjacentes *v*, que estão dentro do raio *r*, não podem ser selecionadas simultaneamente. Observa-se que esta restrição é equivalente à restrição do (2.3), da formulação do PRCP como um PMCIV. A formulação de programação linear inteira do *r*-Separação é definida como se segue:

$$Max \sum x_u \tag{3.19}$$

Sujeito a:

$$x_u + x_v \leq 1 \qquad \forall u = \{1,...,U\} \qquad \forall v \in L_u(r) \tag{3.20}$$

$$x_u, x_v \in \{0,1\} \qquad \forall u = \{1,...,U\} \qquad \forall v \in L_u(r) \tag{3.21}$$

A função objetivo (3.19) visa obter o maior conjunto de vértices, levando em conta a restrição (3.20), de que a distância entre os pontos selecionados deve ser maior que *r* unidades de distância. As restrições (3.4) definem o domínio das variáveis do problema.

Note que quando *r* é maior ou igual à maior distância entre pontos, em uma instância do problema, a formulação do *r*-Separação é equivalente à do PMCIV, pois todos os vértices estão inclusos em $L_u(r)$.

No contexto do PRCP, o objetivo da modelagem como *r*-Separação é garantir uma distância mínima entre os rótulos posicionados. Esta restrição é modelada pela seguinte equação:

$$\sum_{j=1}^{P_i} x_{i,j} \leq 1 \qquad \forall i = \{1,...,N\} \tag{3.22}$$

Também é necessário tentar evitar conflitos ao posicionar os rótulos. Para isto, utiliza-se a variável binária $y_{i,j,k,t}$, em restrições na forma das inequações (2.7), reescritas a seguir.

$$x_{i,j} + x_{k,t} - y_{i,j,k,t} \leq 1, \qquad (k,t) \notin L_{i,j}(r) \tag{3.23}$$

$$(k,t) \in S_{i,j}$$

$$\forall i = \{1,...,N\}$$

$$\forall j = \{1,...,P_i\}$$

Onde $S_{i,j}$ é o conjunto de todas as posições candidatas adjacentes a (i, j) e $L_{i,j}(r)$ é o conjunto de posições candidatas $(k, t) \in S_{i,j}$, que estão a *r* ou mais unidades de distância de (i, j), isto é, $d_{i,j,k,t} \geq r$. Para garantir a distância mínima entre rótulos posicionados, utilizam-se restrições de distância:

$$x_{i,j} + x_{l,m} \leq 1, \qquad (l,m) \in L_{i,j}(r) \tag{3.24}$$

$$\forall\, i = \{1,...,N\}$$

$$\forall j = \{1,...,P_i\}$$

A formulação do PRCP como um problema *r*-Separação é apresentada a seguir:

$$Max \sum x_{i,j} - \sum_{(k,t)\in(S_{i,j}-L_{i,j}(r))} y_{i,j,k,t} \tag{3.25}$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} \leq 1 \qquad \forall i = \{1,...,N\} \tag{3.26}$$

$$x_{i,j} + x_{k,t} - y_{i,j,k,t} \leq 1 \qquad (k,t) \in S_{i,j} - L_{i,j}(r) \tag{3.27}$$

$$\forall i = \{1,...,N\}$$

$$\forall j = \{1,...,P_i\}$$

$$x_{i,j} + x_{l,m} \leq 1 \qquad (l,m) \in L_{i,j}(r) \tag{3.28}$$

$$\forall i = \{1,...,N\}$$

$$\forall j = \{1,...,P_i\}$$

$$x_{i,j}, x_{k,t}, x_{l,m}, y_{i,j,k,t} \in \{0,1\} \qquad (k,t) \in S_{i,j} \tag{3.29}$$

$$(l,m) \in L_{i,j}(r)$$

$$\forall i = \{1,...,N\}$$

$$\forall j = \{1,...,P_i\}$$

A função objetivo (3.25) visa maximizar a quantidade de pontos rotulados e minimizar a quantidade de rótulos posicionados em conflito. Assim como na abordagem do PMCIV, alguns pontos podem não ser rotulados para evitar um conjunto de conflitos na solução.

Observa-se que o modelo como *r*-Separação está relacionado ao *p*-Dispersão. Considerando *U* o conjunto de posições candidatas selecionadas como uma solução viável para uma instância do modelo MD1, o valor de *z* obtido é tal que $d_{i,j,k,t} \geq z, \forall \{(i, j), (k, t)\} \in U$. Então, uma solução viável para o MD1 é também uma solução viável para o modelo *r*-Separação, com $r = z$.

Resultados da abordagem como *r*-Separação são apresentados na Seção 3.4.1 para diferentes valores de *r*.

3.4.1 Resultados do *r*-Separação para o PRCP

Nesta seção são apresentados resultados para o modelo *r*-Separação para as instâncias do conjunto de 5046 pontos, com diferentes dimensões de rótulos e valores de *r*.

Os valores de *r* escolhidos arbitrariamente para os testes são três: 1.14, 9.05 e a maior distância de conflito presente na instância. O tempo de execução dos testes foi limitado em 3000 segundos.

Os resultados para os diferentes valores de raios são apresentados nas colunas com os títulos de $r_{1.14}$, $r_{9.05}$ e r_M, na Tabela 3. As colunas *A* e *L* apresentam as dimensões de rótulos de altura e largura. Os resultados estão agrupados pelos seguintes quesitos de comparação: quantidade de pontos não rotulados (**Pts. não rotulados**), quantidade de rótulos com conflitos (**Rót. com conflitos)**, menor distância de conflito no resultado (***d*_{min}**) e tempo de execução em segundos (**T. execução (*seg*)**).

Tabela 3.5 – Resultados da formulação do PRCP como *r*-Separação.

		Pts. não rotulados			Rót. com conflitos			d_{min}			T. execução (*seg*)		
A	*L*	$r_{1.14}$	$r_{9.05}$	r_M	$r_{1.14}$	$r_{9.05}$	r_M	$r_{1.14}$	$r_{9.05}$	r_M	$r_{1.14}$	$r_{9.05}$	r_M
2	24	9	9	9	0	0	0	-	-	-	20,81	4,03	0,27
2	32	15	15	15	0	0	0	-	-	-	24,91	8,39	0,34
3	16	38	38	38	0	0	0	-	-	-	113,71	8,92	0,45
2	42	27	27	27	0	0	0	-	-	-	57,64	31,79	0,75
2	48	36	39	40	2	8	0	22,02	13,03	-	125,43	43,67	0,79
3	24	83	83	84	2	2	0	14,03	21,02	-	3000	3000	5,60
4	16	108	108	109	2	2	0	2,23	12,16	-	3000	3000	6,77
3	28	105	105	105	0	0	0	-	-	-	3000	3000	2,15
4	18	132	140	140	16	0	0	1,41	-	-	3000	3000	12,54
3	32	127	134	138	8	22	0	2,23	13,03	-	3000	3000	2,95
4	21	174	145	185	26	0	0	1,41	-	-	3000	3000	34,74
4	24	231	213	236	10	2	0	2,23	14,14	-	3000	3000	3000

O modelo garante a distância mínima *r* entre rótulos posicionados em conflito, de forma que $d_{min} \geq r$ para todas as instâncias testadas. Nota-se que conforme o raio aumentou, a quantidade de pontos não rotulados aumentou também, pois com um valor de raio maior, a quantidade de restrições na forma da restrição do PMCIV, as inequações (3.28), também é maior. Em algumas situações, alguns pontos não são rotulados, para que a função objetivo não seja reduzida da quantidade de conflitos que o posicionamento do rótulo geraria. Nas instâncias com o r_M, naturalmente a quantidade de conflitos é 0 em todos os resultados, pois qualquer possível conflito está dentro do raio *r*.

3.5 Considerações Finais

Neste capítulo foi apresentada a modelagem do PRCP como um problema de dispersão discreta. Primeiro este problema foi formalizado através do modelo de programação inteira mista MD1, que teve posteriormente uma reformulação de seu conjunto de restrições de conflitos, no MD2, o que permitiu obter melhores resultados, quando comparados ao modelo FMBPC.

O problema de dispersão discreta está relacionado diretamente ao problema de *r*-Separação e a partir desta abordagem uma outra formulação para o PRCP foi

proposta. A modelagem do PRCP como um problema de *r*-Separação possibilita especificar, com o parâmetro de entrada *r*, uma distância mínima entre rótulos posicionados em conflitos. Isto foi observado em todos os resultados dos testes para este modelo, que foi efetuado variando-se o valor de *r*.

Com o objetivo de levar em conta no problema, não apenas a distância entre os rótulos posicionados, mas também a quantidade de conflitos na solução foi apresentado um terceiro modelo, denominado MD3. Nesta formulação, tenta-se resolver conflitos prioritariamente, e quando não é possível, escolher rótulos com maior valor de distância de conflitos.

Tendo em vista que o MD3 pode gerar um alto custo computacional em instâncias com maior quantidade de conflitos, foi implementado um AGC para obter soluções de instâncias mais complexas. O AGC proposto é apresentado no Capítulo 4 a seguir e seus resultados são comparados com os obtidos do MD3 utilizando o solver CPLEX 12.1.

4 ALGORITMO GENÉTICO CONSTRUTIVO PARA DISPERSÃO DISCRETA NO PRCP

O Algoritmo genético (AG) é uma técnica relacionada aos algoritmos evolutivos, que se tornou popular no campo da Inteligência Artificial no início dos anos 70, mas obteve grande atenção em trabalhos desenvolvidos em áreas diversas, como matemática, engenharia e economia.

Os AGs, assim como outros algoritmos evolutivos, são métodos inspirados na teoria da evolução natural, que utiliza conceitos como mutação, seleção e cruzamento.

Ao modelar um problema de otimização em um AG, é necessário inicialmente definir as possíveis soluções para o problema, em termos de indivíduos em uma população (MITCHELL, 1996). Estes indivíduos, ou soluções, são geralmente representados de maneira análoga aos cromossomos de sistemas biológicos. Estes cromossomos, por sua vez, são compostos por *alelos*, que são elementos parciais da solução do problema a ser modelado. Os valores que cada alelo pode assumir são números inteiros, de ponto flutuante ou caracteres.

Cada solução deve possuir um valor de *fitness* – ou aptidão – associado, que avalia sua qualidade dentro do conjunto de soluções. Existem diversas formas de se determinar este valor (FOGEL, 1998), mas geralmente seu cálculo está associado à função objetivo do problema.

O processo evolutivo do AG inicia-se de um conjunto de soluções iniciais, e iterativamente tenta obter soluções melhores até que uma determinada condição de parada seja alcançada. A cada iteração (geração), novas soluções são geradas, utilizando características dos elementos da população existente, como uma forma de busca na vizinhança destas soluções, de maneira a obter diversidade no algoritmo.

Também, utilizam-se técnicas de intensificação, fazendo alterações aleatórias em poucas soluções, ou aplicando heurísticas de busca local a elas, para escapar de soluções localmente ótimas.

Neste capítulo, é apresentada a implementação de uma variação deste método evolutivo, o Algoritmo Genético Construtivo (AGC), para a resolução do problema de Dispersão Discreta do PRCP. Considerando que a relação de compromisso entre dispersão de rótulos e número de conflitos em uma solução é um tema pertinente de trabalhos futuros, foi utilizado o MD3 apresentado no Capítulo 3, que aborda o problema tanto do ponto de vista da minimização de conflitos quanto da dispersão. O AGC para a Dispersão Discreta do PRCP é apresentado nas seções a seguir.

4.1 Estrutura do AGC para o PRCP

O Algoritmo Genético Construtivo (AGC) é um procedimento iterativo que, assim como o AG tradicional, opera em uma população de soluções onde cada indivíduo representa uma solução diferente para um dado problema. Uma das principais características do AGC é o método para gerar a população inicial, que utiliza o conceito de esquemas, que são soluções incompletas (inviáveis), como base para a construção de soluções melhores para o problema, durante o processo evolutivo. Soluções completas e viáveis são denominadas estruturas.

O uso de esquemas em GAs pode ser datado do trabalho de Holland (1975), que propôs a Hipótese de Blocos de Construção, para formar e conservar esquemas que possam “construir” boas soluções. No entanto, o uso dos blocos de construção necessitava que esquemas fossem avaliados de uma maneira indireta, mais precisamente, através da avaliação das estruturas geradas pelos esquemas. O AGC foi proposto por Lorena e Furtado (2001) como uma alternativa ao GA tradicional, principalmente para permitir que esquemas sejam avaliados diretamente, utilizando uma métrica comum à avaliação das estruturas. Além disto, o AGC utiliza uma função biobjetivo, com um limiar de

rejeição adaptativo, o que resulta em uma população dinâmica durante o processo evolutivo, que inclusive pode ficar vazia.

Com relação ao PRCP, cada alelo de uma solução no AGC corresponde a uma posição candidata de um ponto específico, e um cromossomo corresponde a uma solução para um dado conjunto de pontos. Se um ponto não recebe um rótulo, o alelo correspondente recebe o símbolo # como valor. Por exemplo, considere um conjunto de 8 pontos, com 4 posições candidatas para cada ponto. Um esquema e uma estrutura para este conjunto em particular podem ser ilustrados de acordo com a Figura 4.1.

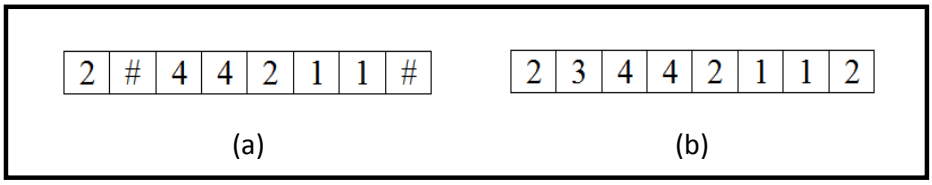

Figura 4.1 – Exemplos de esquema (a) e estrutura (b).

No processo evolutivo do AGC, o primeiro passo consiste de determinar a qualidade dos indivíduos da população de soluções, considerando a função objetivo do problema. Para este propósito utilizam-se duas funções *fitness*: $f(s_k)$ e $g(s_k)$, onde s_k é o k-ésimo indivíduo em uma população. A função f retorna o valor da função objetivo do problema para o indivíduo s_k e g retorna o valor da função objetivo após a aplicação de uma técnica de melhoria da solução, geralmente uma heurística.

O problema original é remodelado como o problema biobjetivo abaixo:

$$\begin{aligned} &\textit{Min } g(s_k) - f(s_k) && (4.1)\\ &\textit{Max } g(s_k)\\ &\textit{Sujeito a:}\\ &\qquad g(s_k) \geq f(s_k) \end{aligned}$$

Ao minimizar a diferença $g(s_k) - f(s_k)$ é necessário maximizar o valor da função objetivo $f(s_k)$, para que se aproxime ao máximo de $max(g(s_k))$. Durante o

processo evolutivo, esta maximização ocorre com a construção de estruturas através da combinação de bons esquemas.

As funções *f* e *g* são utilizadas para atribuir um valor de *ranking* $\delta(s_k)$ para cada indivíduo – estrutura ou esquema – do conjunto de soluções, levando em consideração a "distância" entre a solução e um valor máximo esperado g_{max}, que é um limite superior para $g(s_k)$.

Dado um indivíduo s_k, em que $g(s_k) < g_{max}$ e uma constante $d \in [0, 1]$, o valor de *ranking* $\delta(s_k)$ é definido na seguinte equação.

$$\delta(s_k) = \frac{d \cdot g_{\max} - [g(s_k) - f(s_k)]}{d[g_{\max} - g(s_k)]} \tag{4.2}$$

No AGC proposto neste trabalho para o PRCP, g_{max} é o número total de conflitos na instância, de maneira que se $g(s_k) = g_{max}$, então todos os pontos foram rotulados sem conflitos.

O limiar $\delta(s_k) > \alpha$ é utilizado para aceitar um indivíduo no conjunto de soluções, onde α é um parâmetro de evolução que é iniciado com valor 0 e recebe um incremento a cada nova geração no AGC. Este incremento é feito de acordo com a fórmula

$$\alpha = \alpha + r \cdot |P_\alpha| \cdot \frac{\delta_m - \delta_p}{Gr} + I \tag{4.3}$$

Na expressão acima, *r* é a constante de proporcionalidade, $|P_\alpha|$ é o tamanho da população no tempo de evolução α, δ_m é o melhor *ranking*, δ_p é o pior *ranking*, *Gr*, é a quantidade restante de gerações e *I* é o incremento mínimo de α.

Ao utilizar (4.3), o algoritmo torna-se mais restritivo quanto à aceitação e manutenção de soluções conforme o tamanho da população aumenta, ou conforme o algoritmo se aproxima da quantidade máxima de gerações. De maneira inversa, quando não se tem melhoras significativas nas soluções, α

não tem grandes incrementos, para permitir que indivíduos tenham mais chances de entrar e permanecer na população.

O AGC para o PRCP utiliza a mesma função objetivo do modelo de dispersão apresentado na Seção 3.3, reescrita abaixo.

$$f(s_k) = \sum_{(i,j)\in V(k,t)\in S_{i,j}} \sum \left(\frac{d_{i,j,k,t}}{D} y_{i,j,k,t} \right) + \sum_{(i,j)\in V(k,t)\in S_{i,j}} \sum \left(1 - y_{i,j,k,t}\right) \tag{4.4}$$

D é o máximo $d_{i,j,k,t}$ na instância e $y_{i,j,k,t}$ é a variável indicadora tal que $y_{i,j,k,t} = 1$ se existe um conflito entre (i, j) e (k, t), e $y_{i,j,k,t} = 0$ caso contrário. Assim, a função $f(s_k)$ retorna um valor que leva em conta o número de conflitos e a distância de rótulos em conflito.

A função $g(s_k)$ retorna o valor de (4.4), para uma dada solução s_k, após o uso de um algoritmo de busca local, e para isto, neste trabalho utiliza-se o mesmo método implementado para o operador de Mutação proposto, que será melhor explicado na Seção 4.4. Como a manutenção de bons esquemas é um princípio importante ao utilizar Blocos de Construção, após a aplicação do método de melhoria em s_k a nova função objetiva é armazenada como $g(s_k)$, mas a nova solução é descartada.

No início do AGC a população inicial é composta somente por esquemas com uma proporção predefinida de pontos não rotulados. Outros alelos da solução recebem um número aleatório indicando uma posição candidata para rotular um ponto na forma da padronização de Christensen et al. (1995).

O pseudocódigo do AGC, baseado no algoritmo de Yamamoto e Lorena (2005), é apresentado na Figura 4.2 – Pseudocódigo do AGC. Figura 4.2, para uma população P_α em um tempo de evolução α.

Algoritmo Genético Construtivo

Dados g_{max}, d e α = 0;
Inicialize $P_α$;
para todo *$s_k \in P_α$ calcule $g(s_k)$, $f(s_k)$, $δ(s_k)$;*
enquanto *(não atinge o número máximo de gerações)* ***faça***
    se *α < $δ(s_k)$, adiciona s_k a $P_{α+1}$,* ***para todo*** *$s_k \in P_α$;*
    enquanto *(não atinge número maximo de combinações)* ***faça***
        Selecione s_{base} e s_{guia} em $P_α$;
        Recombine s_{base} e s_{guia} → s_{novo} ;
        Calcule $g(s_{novo})$, $f(s_{novo})$, $δ(s_{novo})$;
        se *α < $δ(s_{novo})$, adicione s_{novo} a $P_{α+1}$;*
    fim_enquanto
    faça *Mutação em s_{base} e mantenha a melhor solução;*
    Atualize α;
fim_enquanto

Figura 4.2 – Pseudocódigo do AGC.

As soluções são ordenadas em $P_α$ do melhor *rank* para o pior, e na operação de seleção, o algoritmo escolhe aleatoriamente uma solução base s_{base} das primeiras soluções em $P_α$. O número de indivíduos nesta sub-população é um parâmetro de entrada do algoritmo. O algoritmo escolhe também uma solução guia s_{guia}, aleatoriamente, dentre todas as soluções em $P_α$.

4.2 Operador de Recombinação

Para a função de Recombinação, o AGC utiliza uma técnica de máscara, proposta por Verner et al. (1997). Esta técnica consiste de mapear valores para cada alelo de um cromossomo, indicando a qualidade local, da rotulação de um ponto em particular, para a solução total. Para a máscara proposta, define-se a função $f_{i,j}$, a contribuição potencial da posição candidata (i, j) para a função objetivo do problema:

$$f_{i,j} = \sum_{(k,t)\in S_{i,j}} \left(\frac{d_{i,j,k,t}}{D} y_{i,j,k,t} \right) + \sum_{(k,t)\in S_{i,j}} \left(1 - y_{i,j,k,t}\right) \quad (4.5)$$

onde D é o máximo $d_{i,j,k,t}$ na instância e $y_{i,j,k,t} = 1$ se existe um conflito entre (i, j) e (k, t), e $y_{i,j,k,t} = 0$ caso contrário. Os valores possíveis de máscara da solução base e solução guia, $M_{base}(i)$ e $M_{guia}(i)$ respectivamente, para cada ponto i são:

- 0, se o ponto não está rotulado (código #);
- $f_{i,j}$, caso contrário.

A nova estrutura ou esquema s_{novo} é obtida aplicando-se a seguinte regra:

Se $M_{base}(i) \geq M_{guia}(i)$ *então*

$s_{novo}(i) \leftarrow s_{base}(i)$;

Senão

$s_{novo}(i) \leftarrow s_{guia}(i)$.

Esta estrutura de seleção assegura que novas soluções recebam o gene de maior valor das máscaras s_{base} e s_{guia}, utilizando o critério da contribuição de um rótulo em particular para o valor total da função objetivo do problema, ao considerar somente seus conflitos evitados e suas distâncias de conflitos não evitados.

4.3 Operador de Mutação

O operador de Mutação aplica uma heurística de busca local para todas as soluções base, e armazena somente a melhor solução mutada. A operação consiste de três passos principais, em que cada passo é uma heurística que tenta melhorar as soluções a serem alteradas, inicialmente com relação ao número de conflitos, e iterativamente as refina para obter melhores valores de dispersão de conflitos.

Sejam C o conjunto de pontos rotulados com conflitos e de pontos não rotulados, em uma dada solução s_{base}, $|C|$ a cardinalidade do conjunto C e

$d_{min}(i, j)$ a menor distância entre a posição candidata (i, j) e suas adjacências em conflito, o pseudocódigo da primeira etapa é apresentado como se segue.

Busca Local 1

Dado s_{base}*;*
Popular C com todos os pontos não rotulados ou rotulados com conflitos em s_{base}*;*
enquanto *|C| > 0*
    Escolher aleatoriamente um ponto i de C;
    Encontrar a posição candidata (i, j) que produz a menor quantidade de conflitos;
    se *houver empate,* ***então***
        Escolher a posição candidata com o maior valor de $d_{min}(i, j)$*;*
    fim_se
    Remover o ponto i de C;
fim_enquanto

Figura 4.3 – Pseudocódigo da Busca Local 1.

A Busca Local 1 procura aleatoriamente, em uma dada solução, por todos os alelos de pontos que não estão rotulados ou estão rotulados com conflitos. Para cada alelo, o método verifica se existe um novo rótulo, que se utilizado resultaria em uma menor quantidade de conflitos para este ponto em particular. Se existir mais do que um rótulo com este mesmo número de conflitos, o algoritmo escolhe o rótulo com a máxima menor distância.

O algoritmo de Busca Local 1 é capaz de encontrar boas soluções de quantidade de rótulos livres de conflitos em um tempo de execução razoável. Porém, utilizar apenas este método pode guiar o AGC para um ótimo local, pois a função objetivo (4.4) leva em conta a distância de conflitos não evitados, além do número de rótulos livres. Tendo em vista este problema, outra heurística é utilizada, como uma segunda etapa da Mutação. Esta etapa tem o objetivo de encontrar valores maiores de distâncias de conflito e consequentemente, possibilitar ao método obter melhores resultados para a

função objetivo. A Figura 4.4 apresenta o pseudocódigo do segundo passo do operador de Mutação.

Busca Local 2

Dado s_{base}*;*
Encontrar d_{min} *em* s_{base}*;*
Popular C com todos os pontos rotulados com conflitos de s_{base}*;*
enquanto $|C| > 0$
    Escolher aleatoriamente um ponto i de C;
    Escolher aleatoriamente uma posição candidata (i, j): $d_{i,j,k,t} > d_{min}$, $\forall (k, t) \in S_{i,j}$;
    Utilizar a posição candidata, se existir, para a rotulação de i ;
    Remover o ponto i de C;
fim_enquanto

Figura 4.4 – Pseudocódigo da Busca Local 2.

A segunda heurística busca aleatoriamente, em uma dada solução, somente por pontos que estão rotulados com conflitos. Para cada um destes pontos, o método verifica se existe um novo rótulo, que se posicionado, poderia aumentar a distância mínima de conflito deste ponto. Esta operação pode introduzir novos conflitos na solução e reduzir o valor da função objetivo. No entanto, os novos conflitos podem dar diversidade ao AGC, o que no longo prazo permite o algoritmo escapar de soluções localmente ótimas. Além disto, como a heurística é aplicada somente nos rótulos em conflito, boa parte da solução é mantida.

No terceiro passo, o objetivo é “reparar” as soluções obtidas da Busca Local 2, e refiná-las na direção da função objetivo. Este refinamento é um procedimento iterativo que busca aleatoriamente por pontos rotulados em conflito, e tenta obter uma rotulação melhor para estes pontos, levando em conta sua contribuição para a função objetivo, avaliada pela equação (4.5). O pseudocódigo do terceiro passo é apresentado na Figura 4.5.

Busca Local 3

Dado s_{base};
Calcular $f_{i,j}$ para todos os pontos em s_{base};
Popular C com todos os pontos de s_{base};
enquanto *|C| > 0*
    Escolher aleatoriamente um ponto i de C;
    Atribuir a i a posição (i, j) com o maior valor de $f_{i,j}$;
    Atualizar $f_{i,j}$;
    Atualizar $f_{k\,t}$ de cada ponto k com o rótulo (k, t) em conflito com (i, j);
    Atualizar $f_{p,q}$ de cada ponto p com o rótulo (p, q) em conflito com o antigo rótulo de i;
    se *o ponto p não possui conflitos,* ***então***
        Remover o ponto p de C
    fim_se
    Remover o ponto i de C;
fim_enquanto

Figura 4.5 – Pseudocódigo da Busca Local 3.

O algoritmo Busca Local 3 tenta encontrar o rótulo *j* com o maior valor de $f_{i,j}$ para o ponto *i*, e atualiza o valor de *f* para todos os pontos com posições candidatas adjacentes, tanto ao rótulo anterior quanto ao novo rótulo de *i.* Este procedimento é então executado para o próximo ponto rotulado com conflitos. Observa-se que pontos rotulados com conflitos, que não foram visitados ainda, podem ter seus conflitos resolvidos durante as trocas de rótulos de outros pontos, e por isto também devem ser removidos de *C*. O pseudocódigo da Mutação é apresentado na Figura 4.6.

Mutação

Dado P_α;
repetir *para todos os indivíduos s_{base} em P_α*
    Executar Busca Local 1 em s_{base};
    Executar Busca Local 2 em s_{base};
    enquanto *(não alcança o número máximo de execuções)* ***faça***
        Executar Busca Local 3 em s_{base};
    fim_enquanto
fim-repetir

Figura 4.6 – Pseudocódigo do operador de Mutação.

Após várias iterações, o terceiro passo perde sua capacidade de melhorar s_{base}, e através de testes computacionais, foi verificado que uma quantidade de 10 iterações é suficiente para obter uma boa melhora e convergência de s_{base}. Após a execução de Busca Local 3, o melhor mutante é armazenado na população e as outras soluções são descartadas.

4.4 Resultados

Esta seção apresenta os resultados dos testes computacionais do MD3 e do AGC proposto. Os resultados do MD3 foram obtidos utilizando o *solver* CPLEX 12.1 (IBM, 2009). As instâncias são as mesmas apresentadas na Seção 3.2.1, que são os conjuntos de 505 e 5046 pontos do mapa da Suíça, considerando 4 posições candidatas para cada ponto, de acordo com a padronização proposta por Christensen et al. (1995). De cada um destes conjuntos, foram geradas 15 instâncias, através da variação das dimensões de altura e largura dos rótulos. A Tabela 4.1 apresenta detalhes das posições candidatas e do número de conflitos das instâncias utilizadas nos testes. As colunas A e L indicam os valores utilizados de altura e largura dos rótulos.

Tabela 4.1 – Número de conflitos em potencial das instâncias de testes.

A	L	Número de conflitos	
		505 Pontos	5046 Pontos
2	24	1660	19120
2	32	2192	24813
3	16	2394	26764
2	42	2659	31181
2	48	2898	34808
3	24	3480	39044
4	16	3831	40484
3	28	3961	44826
4	18	4253	45195
3	32	4452	50337
6	12	4907	50757
4	21	4916	51874
4	24	5437	58489
6	14	5781	59159
6	16	6478	67426

Para a parametrização do AGC foi considerado o trabalho de Yamamoto e Lorena (2005). Outros parâmetros como a proporção de pontos inicialmente rotulados e número máximo de recombinações, foram obtidos empiricamente, através de testes preliminares com diferentes valores. Os parâmetros utilizados são apresentados na Tabela 4.2.

Tabela 4.2 – Parâmetros do AGC.

Parâmetro	505 pontos	5046 pontos
Tamanho da população inicial	30	40
r	0.02	0.015
l	0.00001	0.00001
Quantidade máxima de recombinações	6	6
Quantidade máxima de gerações	200	80
d	0.8	0.5

Proporção de pontos rotulados em esquemas	0.65	0.65
% da sub-população s_{base}	30%	15%

Os resultados do AGC apresentados nesta seção correspondem aos melhores resultados em 10 execuções para cada instância. É utilizada a menor distância entre rótulos em conflito, d_{min}, como uma medida para comparar a dispersão em uma solução. Desta forma, quanto maior o valor de d_{min}, mais esparsos estão os conflitos em uma dada solução. As tabelas Tabela 4.3 e Tabela 4.4 apresentam resultados do modelo de dispersão proposto MD3, resolvido pelo solver comercial CPLEX e os resultados obtidos pelo AGC. O tempo de execução do CPLEX foi limitado em 3600 segundos.

Os resultados apresentados são o tempo de execução, seguido do valor da função objetivo, o *Gap* relativo entre a melhor solução e o limite superior obtidos pelo CPLEX, proporção de rótulos livres de conflitos e a medida de dispersão. As instâncias são ordenadas por sua complexidade, considerando o número de conflitos em potencial. Valores marcados em negrito são os melhores resultados para a proporção de rótulos livres. Traços nos resultados para d_{min} significam que a solução não possui sobreposições de rótulos.

Observa-se que os resultados obtidos são de dois tipos: proporção de rótulos livres de conflitos e a máxima menor distância de conflitos. Estes dois tipos refletem os pontos de vista distintos de rotulação evitando conflitos e dispersão discreta, abordados pelo MD3 e o AGC. No entanto estes resultados não são da função objetivo e foram obtidos através de um pós-processamento, verificando a configuração de rótulos das soluções, para cada instância. Desta forma, a solução ótima do modelo de programação linear inteira pode não ser a solução ótima de rótulos livres de conflitos, nem da dispersão de rótulos.

Tabela 4.3 – Resultados das instâncias de 505 pontos.

505 Pontos										
A	**L**	**T. Exec. (*seg*)**		**Função objetivo**		**Gap**	**Rótulos livres (%)**		d_{min}	
		MD3	**AGC**	**MD3**	**AGC**	**(%)**	**MD3**	**AGC**	**MD3**	**AGC**
2	24	0.1	0.2	1660.0	1660.0	0.0	100.0	100.0	-	-
2	32	0.1	0.2	2192.0	2192.0	0.0	100.0	100.0	-	-
3	16	0.2	0.4	2394.0	2394.0	0.0	100.0	100.0	-	-
2	42	0.6	115.8	2658.7	2658.7	0.0	99.2	99.2	33.0	33.0
2	48	1.2	220.0	2897.6	2897.6	0.0	99.2	99.2	39.0	39.0
3	24	3.7	225.0	3479.3	3479.3	0.0	98.4	98.4	14.0	14.0
4	16	4.1	224.8	3829.5	3829.5	0.0	96.8	96.8	9.2	9.2
3	28	25.3	225.0	3960.0	3960.0	0.0	97.2	97.2	**18.0**	16.8
4	18	81.6	230.9	4251.1	4251.1	0.0	95.4	95.4	8.5	8.5
3	32	4.5	236.5	4449.9	4449.9	0.0	95.8	95.8	11.1	11.1
6	12	19.7	233.4	**4902.2**	4901.7	0.0	92.4	92.4	**6.7**	4.2
4	21	12.8	240.8	4911.9	4911.9	0.0	93.2	93.2	8.8	8.8
4	24	43.2	240.2	5431.0	5431.0	0.0	90.2	90.2	8.6	8.6
6	14	687.5	244.3	5772.3	5772.3	0.0	**87.3**	85.3	2.2	**4.2**
6	16	828.7	255.8	**6464.7**	6462.3	0.0	82.1	82.1	**4.0**	2.2

Tabela 4.4 – Resultados das instâncias de 5046 pontos.

		5046 Pontos								
A	**L**	**T. Exec. (*seg*)**		**Função objetivo**		**Gap**	**Rótulos livres (%)**		d_{min}	
		MD3	**AGC**	**MD3**	**AGC**	**(%)**	**MD3**	**AGC**	**MD3**	**AGC**
2	24	10.3	1030.3	19119.0	19119.0	0.0	99.6	99.6	9.0	9.0
2	32	82.9	1031.5	24810.4	24810.4	0.0	99.2	99.2	15.0	15.0
3	16	66.7	1030.4	26758.2	26758.2	0.0	98.2	98.2	3.1	3.1
2	42	109.8	1025.8	31174.6	31174.6	0.0	98.6	98.6	8.0	8.0
2	48	166.3	1033.5	34798.8	34798.8	0.0	98.0	98.0	14.0	14.0
3	24	835.8	1048.7	**39019.0**	39017.8	0.0	95.6	**96.4**	**4.8**	2.2
4	16	3609.5	1062.1	**40423.0**	40419.4	30.0	**95.6**	95.0	**4.1**	4.0
3	28	3666.9	1060.2	**44790.3**	44785.1	10.0	**94.8**	94.2	2.2	2.2
4	18	3588.0	1058.0	45134.0	45134.0	30.0	91.2	91.2	1.4	1.4
3	32	3637.0	1068.7	50283.0	50283.0	16.0	89.2	89.2	4.4	4.4
6	12	3658.9	1070.7	50654.9	**50655.1**	32.9	88.8	**89.6**	2.2	2.2
4	21	3648.4	1077.7	51761.8	**51763.0**	48.1	89.9	89.9	2.2	**3.1**
4	24	3605.6	1110.4	58363.4	**58366.2**	39.3	85.0	**86.4**	2.2	2.2
6	14	3598.2	1085.6	58953.3	**58975.6**	57.5	83.0	**84.5**	1.4	2.2
6	16	3598.2	1101.1	67140.7	**67161.5**	59.4	79.1	**83.4**	1.4	2.2

O AGC obteve o mesmo resultado do CPLEX para a maioria das instâncias de 505 pontos, e melhores soluções de dispersão e rótulos livres para várias das instâncias mais complexas de 5046 pontos. O modelo de programação inteira de dispersão apresentou melhores resultados que o AGC para o tempo de execução dos testes, em instâncias menores, e provou a solução ótima para todas as instâncias de 505 pontos. No entanto, conforme o número de conflitos em potencial aumenta nas instâncias maiores (5046 pontos), o modelo de dispersão tem o desempenho piorado, por causa do aumento da quantidade de variáveis de decisão e restrições, já que para cada possível conflito existe uma restrição correspondente na forma da inequação (3.16). Por outro lado, o tamanho de um cromossomo do AGC depende somente do número de pontos, o que permite um tempo de execução mais estável.

Para este modelo proposto, é possível obter soluções com o mesmo resultado de função objetivo, mas com diferentes valores de proporção de rótulos livres e

de dispersão. É igualmente possível ter um valor pior na função objetivo, mas com um bom resultado de rótulos livres de conflitos. Isto pode ser observado em resultados da instância de 5046 pontos, A=3 e L=24, em que a função objetivo do modelo de programação inteira é melhor, mas tem pior resultado de proporção de rótulos livres.

4.5 Considerações finais

Este capítulo apresentou um Algoritmo Genético Construtivo, com o objetivo de de obter o menor número de conflitos de rótulos, e obter a máxima menor distância entre rótulos, quando conflitos não podem ser evitados, como uma nova abordagem para o PRCP. Para isto, utilizou-se o conceito de dispersão discreta, apresentado no Capítulo 3.

Os resultados do AGC foram comparados aos obtidos pelo solver CPLEX 12.1, resolvendo instâncias do MD3. O solver obteve a solução ótima de todas as instâncias de 505 pontos e de algumas das de 5046 pontos. No entanto, para instâncias mais complexas, o AGC obteve os melhores resultados para a função objetivo e o tempo de execução.

Os resultados obtidos pelo AGC indicam um bom passo inicial para resolver instâncias com grande quantidade de pontos, em trabalhos futuros. Outro método de resolução de problemas em instâncias de larga escala é o de decompor em problemas menores estas instâncias, de maneira que a resolução destes subproblemas apresente uma boa solução para o problema original.

Uma técnica de decomposição que apresentou bons resultados na literatura para o PRCP é a Decomposição Lagrangeana proposta para o PMNRSC de Mauri et al. (2010). Tendo em vista os bons resultados, no Capítulo 5 a seguir apresenta-se uma investigação sobre esta técnica, aplicada ao PRCP e uma abordagem desbanceada da Decomposição Lagrangeana é proposta. Comparação de resultados computacionais entre a versão tradicional e a versão desbalanceada também são apresentados.

5 DECOMPOSIÇÃO LAGRANGEANA DESBALANCEADA PARA O PRCP

Neste Capítulo, apresentam-se a abordagem de Decomposição Lagrangeana (DecLag) desbalanceada e os resultados obtidos para instâncias de testes do PRCP. É considerado o modelo proposto por Mauri et al. (2010), que apresentou soluções ótimas para diversas destas instâncias. Porém, o método não conseguiu comprovar a otimalidade de diversas instâncias do conjunto de 1000 pontos, pois para estas instâncias, o Subgradientes (NARCISO; LORENA, 1999) teve dificuldade em obter bons limitantes para as soluções.

A DecLag, apresentada por Guignard e Kim (1987), permite a decomposição de um problema em diversos subproblemas que, ao serem solucionados de maneira independente, obtém-se a solução do problema original, sem perda de generalidade.

Para a decomposição, o grafo de conflitos do problema foi inicialmente modelado como um problema de particionamento. As partições obtidas são consideradas como subproblemas do problema original. Porém, ao particionar um grafo, é possível que vértices adjacentes fiquem em partições diferentes, gerando arestas entre partições. Esta quantidade de arestas interpartições é denominada na literatura de *corte* de arestas.

Como as arestas entre vértices do grafo de conflitos para o PRCP indicam um conflito entre vértices, elas devem ser consideradas também. Neste caso, um vértice que possua adjacência externa é copiado para a partição do vértice adjacente, de maneira que sua restrição de conflito seja considerada em um dos subproblemas.

Para que as características do problema original sejam mantidas, são necessárias restrições adicionais, que assegurem que os valores das variáveis copiadas sejam iguais aos valores das variáveis originais. Estas restrições são então relaxadas no sentido Lagrangeano (GUIGNARD, 2003), permitindo que os subproblemas sejam resolvidos de maneira independente, com um

problema dual Lagrangeano correspondente, a ser otimizado utilizando o método de Subgradientes.

Porém, o tempo de convergência e o *gap* dos limites obtidos, ao resolver a relaxação lagrangeana, pelo método de Subgradientes, é sensível à quantidade de restrições relaxadas, de maneira que quanto maior esta quantidade, mais difícil é obter um bom limitante. Tendo isto em vista, e considerando que o particionamento desbalanceado pode permitir um melhor valor para o corte, reduzindo consequentemente a quantidade de restrições relaxadas, é proposto neste trabalho uma Decomposição Lagrangeana de maneira desbalanceada, para obter melhores resultados para o problema dual e eventualmente a solução ótima de instâncias do PRCP.

O modelo de Decomposição Lagrangeana para o PRCP, o algoritmo de particionamento desbalanceado e os resultados obtidos são discutidos nas seções a seguir.

5.1 Decomposição Lagrangeana para o PMNRSC

O modelo proposto por Mauri et al. (2010) leva em conta a abordagem de PMNRSC, com os rótulos tendo dimensões fixas e utilizando padronização cartográfica de Christensen et al. (1995). Utiliza-se a variável de decisão binária $x_{i,j}$ como nos modelos anteriores, para identificar se a posição candidata (i, j) é utilizada ou não, e um peso $w_{i,j}$ indicando a recompensa por escolher a posição candidata (i, j). O modelo de programação inteira para o PMNRSC é definido por

$$Max \sum_{i=1}^{N} \sum_{j \in P_i} w_{i,j} x_{i,j} - \sum_{i=1}^{N} z_i \tag{5.1}$$

Sujeito a:

$$\sum_{j \in P_i} x_{i,j} = 1 \qquad \forall\, i = \{1,\ldots,N\} \tag{5.2}$$

$$x_{i,j} + x_{t,u} - z_i \leq 1 \qquad \forall\, i = \{1,...,N\} \qquad (5.3)$$

$$\forall\, j = \{1,...,P_i\}$$

$$(t,u) \in S_{i,j}$$

$$x_{i,j}, x_{t,u}, z_i \in \{0,1\} \qquad \forall\, i = 1,...,N \qquad (5.4)$$

$$\forall\, j = 1,...,P_i$$

$$(t,u) \in S_{i,j}$$

onde $z_i = 1$ indica que o ponto i foi rotulado com conflito, $z_i = 0$ representa o caso contrário. A restrição (5.2) assegura que cada ponto seja rotulado, e que tenha um único rótulo. A restrição (5.3) atribui os valores para as variáveis z_i caso exista um conflito potencial entre um dado par de vértices $x_{i,j}$ e $x_{t,u}$. Restrição (5.4) indica as variáveis binárias do problema.

Na tentativa de maximizar a função objetivo (5.1), é importante obter a maior quantidade possível de variáveis z_i com valor zero, pois penalizam o somatório duplo na função.

A resolução da formulação (5.1) – (5.4) pode ser computacionalmente muito custosa, e por isso foi proposta por Mauri et al. (2010) uma Decomposição Lagrangeana, para dividi-la em subproblemas e resolvê-la independentemente.

Inicialmente o grafo G é particionado em k partições, em que T_m é o subconjunto de vértices de G, representando a partição $m = \{1,...,k\}$. Em seguida, são selecionados vértices que possuem adjacências externas, para serem copiados para uma das partições de seus vértices adjacentes. Para isto, são definidos os conjuntos C_m, de vértices copiados para a partição T_m e $X_m = V - T_m$, o conjunto de todos os vértices que não estão em T_m.

Para obter as partições T_m foi utilizada a heurística Metis (KARYPIS; KUMAR, 1998), resolvendo o problema de k-Particionamentos, onde k é a quantidade de partições desejada. A forma geral do k-Particionamentos é obter k partições de

maneira que cada partição contenha aproximadamente a mesma quantidade de vértices.

Para obter os conjuntos C_m, utiliza-se a estratégia proposta por Sachdeva (2004), que iterativamente seleciona o vértice com a maior quantidade de arestas interpartições e o copia para o cluster com a maior incidência de arestas interpartições. Ao fazer isto, estas arestas de conflito entre partições passam a não existir mais, e as restrições de conflito referentes a elas são usadas no subproblema da partição onde o vértice foi copiado. Em seguida este número de arestas é redefinido e o procedimento é repetido.

Dadas as partições T_m e os conjuntos C_m e X_m, o PMNRSC é remodelado:

$$Max \sum_{m=1}^{k} \left(\sum_{(i,j)\in T_m} w_{i,j} x_{i,j} - \sum_{(i,j)\in T_m} z_i \right) \tag{5.5}$$

Sujeito a:

$$\sum_{j\in P_i} x_{i,j} = 1 \qquad \forall i \in T_m \qquad m = \{1,...,k\} \tag{5.6}$$

$$x_{i,j} + x_{t,u} - z_i \leq 1 \qquad \forall (i,j) \in T_m \qquad \forall (t,u) \in S_{i,j} \qquad m = \{1,...,k\} \tag{5.7}$$

$$x_{i,j} + x^m_{t,u} - z_i \leq 1 \qquad \forall (i,j) \in T_m \qquad \forall (t,u) \in C_m \cap S_{i,j} \qquad m = \{1,...,k\} \tag{5.8}$$

$$x_{i,j} + x^m_{t,u} - z^m_t \leq 1 \qquad \forall (i,j) \in T_m \qquad \forall (t,u) \in C_m \cap S_{i,j} \tag{5.9}$$

$$m = \{1,\ldots,k\}$$

$$x_{t,u}^{m} = x_{t,u} \qquad \forall (t,u) \in C_m \cap S_{i,j} \qquad (5.10)$$

$$m = \{1,\ldots,k\}$$

$$z_{t}^{m} = z_{t} \qquad \forall (t,u) \in C_m \cap X_m \qquad (5.11)$$

$$m = \{1,\ldots,k\}$$

$$x_{i,j}, x_{t,u}, x_{t,u}^{m}, z_i, z_t, z_t^m \in \{0,1\} \qquad \forall (i,j) \in T_m \qquad (5.12)$$

$$\forall (t,u) \in T_m \cup C_m$$

$$m = \{1,\ldots,k\}$$

onde a variável $x_{t,u}^{m}$ representa a posição candidata (*t, u*) copiada na partição *m* e z_t^m é a variável indicadora de rotulação com conflitos, de um ponto *t* que teve suas posições candidatas copiadas em *m*. A restrição (5.8) considera os conflitos interpartições, e determina se um ponto externo à partição *m* é rotulado com conflitos. A restrição (5.9) considera os conflitos interpartições determinando se um ponto da partição *m* é rotulado com conflitos. As restrições (5.10) e (5.11) asseguram a igualdade entre as variáveis de vértices copiados e as variáveis dos vértices originais.

Ao relaxar as restrições (5.10) e (5.11) sentido Lagrangeano, cada partição pode ser modelada em um subproblema v_m como se segue, $\forall\ m = \{1,\ldots, k\}$:

$$v_m = Max \qquad (5.13)$$

$$\sum_{(i,j)\in T_m} \left(w_{i,j} - \sum_{d \neq m} \alpha_{i,j}^{d} \right) x_{i,j} + \sum_{(t,u)\in C_m} \left(\alpha_{t,u}^{m}\right) x_{t,u}^{m} - \sum_{(i,j)\in T_m} \left(z_i - \sum_{d \neq m} \beta_i^d \right) z_i + \sum_{(t,u)\in C_m} \left(\beta_t^m\right) z_t^m$$

Sujeito a:

$$\sum_{j \in P_i} x_{i,j} = 1 \qquad \forall i \in T_m \qquad (5.14)$$

$$m = \{1,...,k\}$$

$$x_{i,j} + x_{t,u} - z_i \leq 1 \qquad \forall (i,j) \in T_m \qquad (5.15)$$

$$\forall (t,u) \in S_{i,j}$$

$$m = \{1,...,k\}$$

$$x_{i,j} + x_{t,u}^m - z_i \leq 1 \qquad \forall (i,j) \in T_m \qquad (5.16)$$

$$\forall (t,u) \in C_m \cap S_{i,j}$$

$$m = \{1,...,k\}$$

$$x_{i,j} + x_{t,u}^m - z_t^m \leq 1 \qquad \forall (i,j) \in T_m \qquad (5.17)$$

$$\forall (t,u) \in C_m \cap S_{i,j}$$

$$m = \{1,...,k\}$$

$$x_{i,j}, x_{t,u}, x_{t,u}^m, z_i, z_t, z_t^m \in \{0,1\} \qquad \forall (i,j) \in T_m \qquad (5.18)$$

$$\forall (t,u) \in T_m \cup C_m$$

$$m = \{1,...,k\}$$

onde α e β são os multiplicadores Lagrangeanos, variáveis do seguinte problema dual a ser otimizado:

$$Min \sum_{m=1}^{k} v_m \qquad (5.19)$$

Sujeito a:

$$\alpha, \beta \geq 0 \qquad (5.20)$$

É necessário ainda garantir soluções viáveis para o problema, já que restrições do problema original podem ser violadas ao resolver o modelo (5.19) – (5.20). Para isto é utilizada uma heurística gulosa, que busca a melhor alteração da

posição candidata para cada ponto, atribuindo o valor 1 para a variável correspondente à posição escolhida e 0 para as demais, assegurando a viabilidade da solução. A cada alteração de valor das variáveis, a função objetivo do modelo original é recalculada, e a melhor solução é armazenada.

Como mencionado anteriormente, quanto maior a quantidade de restrições (5.10) e (5.11) relaxadas, mais complicado é encontrar bons limitantes duais para o problema decomposto, de forma que obter bons resultados de particionamento podem facilitar a obtenção da solução ótima para uma dada instância do problema.

Considerando um particionamento balanceado, quanto maior o número k de partições, maior tende ser o valor do corte, com um pior caso de $k = N$, onde o corte é a quantidade de arestas no problema. Por outro lado, um número pequeno de partições, que reduza o valor do corte não necessariamente é uma boa opção, pois o tamanho dos subproblemas pode ser grande demais para ser resolvido por um método exato como o B&B.

Uma alternativa é permitir que o tamanho das partições seja inomogêneo, de maneira que a inclusão de vértices em uma partição, que reduza a quantidade de arestas entre partições, não seja limitada pelo fator de balanceamento. Isto pode ser usado para obter uma maior quantidade de subproblemas, mas com bons resultados de corte de arestas. O método proposto para obter as partições T_m de maneira desbalanceada é apresentado na seção a seguir.

5.2 Particionamento desbalanceado

O problema de particionamento geral, conhecido como particionamento em k-vias, ou k-Particionamento, é o problema de obter para um determinado grafo $G = (V, A)$, k subconjuntos disjuntos de vértices, de maneira que o peso total das arestas entre partições, ou corte de arestas, seja minimizado, e o número de vértices em cada subconjunto seja aproximadamente o mesmo. Sabe-se que a resolução para este problema, mesmo com $k = 2$, é NP-Difícil (BUI; JONES, 1993).

Do ponto de vista da otimização combinatória, a minimização do corte é considerada a função objetivo do problema e o balanceamento entre as partições é uma restrição. No entanto existem algumas variações desta função objetivo, que consideram, por exemplo, minimizar especificamente o número de partições interconectadas. Uma variação do problema de particionamento, utilizada neste trabalho é a de permitir o desbalanceamento dos clusters para permitir uma redução da quantidade de arestas entre partições e se possível, o número de partições interconectadas.

Um resultado de particionamento é dito ser balanceado se, $\forall m = \{1,...,k\} : |T_m| \leq L = \lceil |V| / k \rceil$, onde $|T_m|$ é a cardinalidade da partição T_m. Uma solução em que $|T_m| = L$, $\forall\ m = \{1,..., k\}$, é considerada *perfeitamente balanceada*.

O algoritmo para particionamento desbalanceado apresentado nesta seção considera um dado fator máximo de balanceamento F_{bal}, de maneira que a inequação $|T_m| / L \leq F_{bal}, \forall m = \{1,...,k\}$ seja a nova restrição de balanceamento utilizada, com $F_{bal} > 1$. O algoritmo proposto é apresentado na próxima Seção.

5.2.1 Algoritmo de particionamento desbalanceado

Nesta Seção apresenta-se o algoritmo para particionamento desbalanceado para obter os subproblemas para a Decomposição Lagrangeana. O algoritmo tem como base o trabalho de Hendrickson e Leland (1995), que apresentou uma heurística multinível para o *k*-Particionamento e é atualmente o arcabouço mais utilizado para a resolução de problemas de particionamento em geral.

O algoritmo tem como passo principal o colapsamento iterativo de vértices, coalescendo pares de nós adjacentes, gerando um grafo menor ao final de cada iteração. Cada novo grafo obtido é dito ser de nível superior ao do grafo anterior, e o grafo original é chamado de grafo de nível zero. O método é aplicado recursivamente, para formar grupos de vértices, que são utilizados para formar uma partição inicial do problema.

Após o particionamento inicial, o método retorna iterativamente ao grafo original, separando os vértices colapsados e aplicando um algoritmo de refinamento da solução de particionamento em cada iteração.

Para o método apresentado nesta seção, são definidos os conjuntos V_l, de vértices de G no nível l, com $V_0 = V$. É interessante, no particionamento para a DecLag do PMNRSC, que todas as posições candidatas (i, j) de um mesmo ponto sejam coalescidas em um mesmo subconjunto C_i no nível um ($l = 1$), para cada ponto i. Isto porque estes vértices formam cliques, cujas arestas já estarão inclusas neste coalescimento. Então $V_1 = \{C_1, C_2,...C_N\}$ e $C_i = \{(i, 1), (i, 2),...(i, P_i)\}$, $\forall\ i = \{1,...,N\}$, onde N é o número de pontos na instância e P_i o número de posições candidatas de i.

Adicionalmente, definem se as arestas $a(C_i, C_s) \in A_l$, indicando adjacência entre os subconjuntos C_i e C_s, se existir adjacência entre ao menos um par de vértices (i, j) e (s, t), $\forall\ (i, j) \in C_i$ e $\forall\ (s, t) \in C_s$; e os pesos $w_{a(C_i,C_s)}$ associados a estas arestas. O algoritmo da etapa de coalescimento é apresentado no pseudocódigo da Figura 5.1.

A partir de $l = 1$, dadas as arestas entre cada elemento de V_l, seleciona-se aleatoriamente um elemento C_i. Seleciona-se também o elemento adjacente C_s, que possui a menor cardinalidade. Como inicialmente $|C_s| = P_s\ \forall\ C_s \in V_l$, isto é, todos os conjuntos C_s possuem a mesma quantidade de vértices, que são as posições candidatas de cada ponto s, o primeiro elemento visitado será o escolhido.

Dados os elementos selecionados no passo anterior, para o próximo nível do grafo, todos os vértices de C_s são adicionados a C_i, e C_s é removido. Desta maneira, a cardinalidade de V_{l+1} é reduzida de um, mas a cardinalidade de C_i é incrementada de $|C_s|$. Os elementos C_i e C_s são marcados para que não sejam selecionados novamente nesta iteração e o procedimento é reiniciado. Este processo é repetido enquanto houverem elementos a serem selecionados e ao finalizar, l é incrementado.

Coalesce Vértices

$V_0 = V$;
Adicionar (i, j) em C_i, $\forall i = \{1,...,N\}$ e $j = \{1,..., P_i\}$;
$l = 1$;
$V_l = \{C_1, ..., C_N\}$;
enquanto $|V_l| < |V_{l-1}|$ *ou* $|V_l| > k$
    $V_{l+1} = V_l$;
    enquanto *(houver* C_i *não selecionado em* V_{l+1}*)*
        encontrar aleatoriamente C_i *em* V_{l+1}*, que não tenha sido selecionado;*
        marcar C_i *como selecionado;*
        encontrar C_s *em* V_{l+1} *tal que* $|C_s|$ seja a menor, $\forall C_s \in V_{l+1}$ *e*
        $\exists a(C_i, C_s) \in A$ *e* C_s *não tenha sido selecionado;*
        se $\exists \{C_i, C_s\}$ ***então***
            marcar C_s *como selecionado;*
            $C_i = C_i \cup C_s$;
            $V_{l+1} = V_{l+1} - C_s$;
        fim_se
    fim_enquanto
    $l = l + 1$;
fim_enquanto

Figura 5.1 – Pseudocódigo do algoritmo Coalesce Vértices.

Se V_l possuir uma cardinalidade menor que V_{l-1}, o algoritmo fará mais uma iteração dos passos anteriores na tentativa de reduzir o grafo novamente. Caso contrário, o algoritmo finaliza. Outro critério de parada é a cardinalidade de V_l ser igual à quantidade *k* de partições.

Como resultado da execução do algoritmo Coalesce Vértices, o grafo estará reduzido a um conjunto menor de vértices, facilitando assim, determinar uma partição inicial de *G,* obtida pelo algoritmo Obtém Partição Inicial, apresentado na Figura 5.2 a seguir.

Obtém Partição Inicial

repetir *para todo* $C_i \in V_l$

encontrar C_i *que possui a menor* $|C_i|$, $\forall\ C_i \in V_l$;

encontrar a partição T_m *tal que* $T_m = T_m \cup C_i$ *produza a menor violação de* F_{bal};

se $|T_m \cup C_i|/L \leq F_{bal}$ ***então***

$T_m = T_m \cup C_i$;

fim_se

senão

executar ***Obtém Partição Inicial*** *em* V_{l-1};

fim_senão

fim_repetir

Figura 5.2 – Pseudocódigo do algoritmo Obtém Partição Inicial.

No algoritmo da Figura 5.2, l é iniciado do último incremento Coalesce Vértices, ou seja, inicia-se do último nível de coalescimento obtido para G, e sendo assim, V_l possui poucos subconjuntos C_i, que por sua vez contêm grandes quantidades de vértices do grafo original. Nesta etapa, o objetivo é obter uma partição inicial para G, inserindo os elementos de V_l nas partições T_m, $\forall\ m \in \{1,..., k\}$, levando em conta o fator de balanceamento F_{bal} desejado. Se eventualmente não é possível selecionar um elemento C_i que não viole F_{bal}, o algoritmo é aplicado recursivamente em V_{l-1}.

Os elementos de C_i, coalescidos em V_l, estão separados em V_{l-1}, e consequentemente possuem uma cardinalidade menor. O algoritmo efetua a busca novamente por um conjunto C_i que não viole F_{bal} e se encontrado, ele é inserido na partição T_m tal que $|T_m \cup C_i|/L$, $\forall\ m \in \{1, ..., k\}$.

Apesar de o algoritmo Obtém Partição Inicial determinar uma partição para G, ele não considera o problema de minimizar a quantidade de arestas entre as partições. Para isto, neste trabalho, é utilizado um algoritmo baseado na

técnica proposta por Kernighan e Lin (1970). Esta técnica utiliza o conceito de "matriz de ganho", que contém para cada vértice do grafo, valores indicando a contribuição para a minimização do corte ao "mover" este vértice da partição atual para outras partições. Para isto, utiliza-se o valor de ganho $g^m(C_i)$ ao mover o elemento C_i de sua partição atual p para a partição m. Dados os pesos $w_{a(C_i,C_s)}$ para cada aresta $a(C_i, C_s) \in A_l$, este valor pode ser definido como

$$g^m(C_i) = \sum_{a(C_i,C_s)\in A_l} \begin{cases} w_{a(C_i,C_s)} \; se \; C_s \in T_m \\ -w_{a(C_i,C_s)} \; se \; C_s \in T_p \\ 0 \quad se \; C_s \in T_r, r \neq p, r \neq m \end{cases} \tag{5.21}$$

No somatório em (5.21), o ganho em mover C_i para T_m aumenta para cada elemento adjacente C_j que estiver na nova partição. Se C_s estiver na partição atual T_p, uma penalidade é imposta ao valor de $g^m(C_i)$, e se C_s não pertencer nem a T_m e nem a T_p, não existe alteração no ganho. O valor de $g^m(C_i)$ é utilizado no pseudocódigo a seguir.

Refina Particionamento

enquanto *(não alcança a condição de parada)*

Calcular $g^m(C_i)$ $\forall C_i \in V_l$ *que não tenha sido movido*, $\forall$ $m = \{1, ..., k\}$;

Obter o maior valor de $g^m(C_i)$, tal que $|T_m \cup C_i| / L \leq F_{bal}$;

Dado $g^m(C_i)$, *mover o conjunto* C_i *para a partição m;*

Marcar i como "movido";

fim_enquanto

Figura 5.3 – Pseudocódigo do algoritmo Refina Particionamento.

No refinamento da solução, o algoritmo busca pelos melhores movimentos de vértices utilizando os maiores valores de ganho $g^m(C_i)$. O procedimento reitera enquanto houverem vértices não selecionados que tenham um movimento possível que melhore o resultado do corte, e não viole F_{bal}. A Figura 5.4 ilustra o funcionamento do algoritmo multinível com $k = 3$.

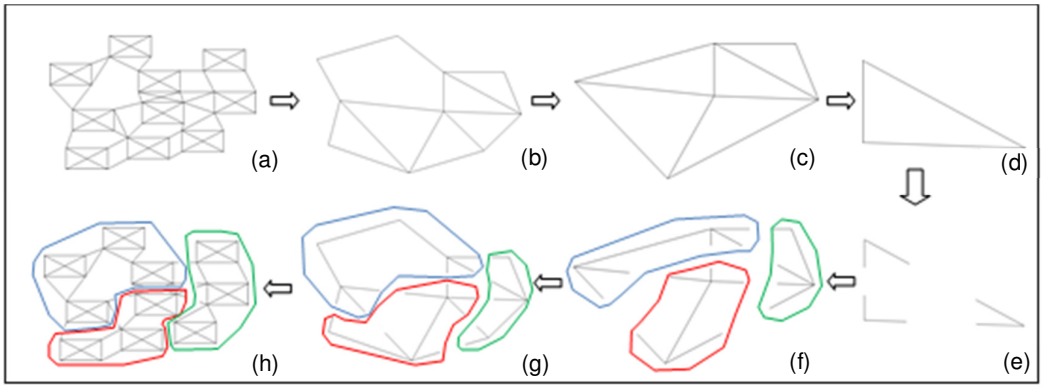

Figura 5.4 – Exemplo de particionamento multinível com $k = 3$.

Na Figura 5.4, inicialmente, as posições candidatas de cada ponto são coalescidas em um único vértice em (a). Em seguida, pares destes vértices são coalescidos também, e o processo é repetido até que a quantidade de vértices no grafo é igual ao valor $k = 3$ ((b) – (d)). A partir desta solução, o grafo é inicialmente particionado em (e) e expandido utilizando o método de refinamento ((f) – (h)).

5.3 Resultados

Nesta seção são apresentados resultados computacionais da decomposição desbalanceada do PMNRSC para as instâncias 1000 pontos, que foram utilizadas para os testes de Mauri et al. (2010) e não tiveram as soluções ótimas comprovadas, considerando quatro posições candidatas para cada ponto, de acordo com a padronização de Christensen et al. (1995). Estas instâncias, geradas no trabalho de Yamamoto et al. (2002), estão disponíveis no site http://www.lac.inpe.br/_lorena/instancias.html.

Primeiro, são apresentados na Tabela 5.1, os resultados de testes com as mesmas quantidades de partições, utilizadas no trabalho de Mauri et al. (2010). A quantidade de partições é apresentada na coluna com o cabeçalho *k*, seguido das melhores soluções obtidas na coluna LI, e dos limitantes duais na coluna LS. Em seguida é apresentado o *Gap* relativo entre LI e LS, e os tempos de execução do método. Os testes foram executados utilizando o

solver comercial CPLEX 12.6, utilizando o fator $F_{bal} = 2$, para a decomposição desbalanceada. Ou seja, $\forall m = \{1,...,k\} : |T_m| / L \leq 2$.

Como esperado, a Tabela 5.1 mostra melhores resultados para o limite dual obtido pela versão desbalanceada da Decomposição Lagrangeana, possibilitando provar a otimalidade de 13 das 19 instâncias teste. Além disto, foram obtidos novos resultados para a função objetivo do problema em quatro instâncias. Foi possível também obter melhores resultados de tempos de execução do método para 12 instâncias. Em todos os casos, o *Gap* da decomposição desbalanceada foi melhor.

Tabela 5.1 – Resultados utilizando o particionamento de Mauri et al. (2010).

		Desbalanceado				**Balanceado**			
Instância	***k***	**LI**	**LS**	***Gap***	**Tempo de execução (*s*)**	**LI**	**LS**	***Gap***	**Tempo de execução (*s*)**
1000_2	25	934	935.5	**0.16**	47189.06	934	962.59	2.97	**9372.64**
1000_4	40	933	935.52	**0.26**	14703.38	933	962.43	3.05	**9234.03**
1000_5	15	961	962	**0.1**	12194.26	961	976.20	1.55	**8834.52**
1000_7	25	**929**	929.94	**0.1**	**1927.41**	928	931.99	0.42	14419.30
1000_8	20	940	940	**0**	**1561.10**	940	942.21	0.23	14419.89
1000_9	20	**927**	927.92	**0.09**	**4728.74**	925	959.52	3.59	8894.97
1000_10	20	944	944.89	**0.09**	**5483.18**	944	971.54	2.83	8861.16
1000_12	25	**935**	935.99	**0.1**	**1443.80**	934	936.44	0.26	14562.00
1000_13	25	954	955.20	**0.12**	18582.80	**955**	972.99	1.84	**8964.73**
1000_14	25	933	934.17	**0.12**	19250.06	933	960.52	2.86	**9293.41**
1000_15	25	934	934.31	**0.03**	75274.17	934	963.99	3.11	**9248.70**
1000_16	25	**932**	932.97	**0.1**	**3030.37**	931	959.58	2.97	9647.72
1000_17	25	937	937	**0**	**1269.61**	937	964.61	2.86	11243.55
1000_18	25	946	946.97	**0.1**	**1132.63**	946	947.27	0.13	14509.09
1000_19	25	950	950.98	**0.1**	**2969.85**	950	953.25	0.34	15264.32
1000_21	25	930	930	**0**	**177.14**	930	955.38	2.65	8707.73
1000_22	25	952	952.99	**0.1**	**2377.17**	952	955.41	0.35	14406.28
1000_23	25	934	934.97	**0.1**	**3454.19**	934	960.13	2.72	9103.09
1000_24	25	931	933.01	**0.21**	12721.37	**932**	964.74	3.39	**9178.91**

Além dos testes com as quantidades de partições utilizadas por Mauri et al. (2010), foram executados testes com outros valores de *k*. Para a escolha destas partições, foi adotado principalmente o critério do menor valor de corte gerado no particionamento.

Adicionalmente, foi observado que para outros valores de F_{bal}, é possível obter quantidades de variáveis copiadas e restrições relaxadas similares ao valor do corte, implicando em uma menor quantidade de arestas interpartições incidentes nos vértices a serem copiados. Isto por sua vez significa que uma dada variável copiada é utilizada em poucas restrições de conflitos dos tipos (5.16) e (5.17), por precisarem eliminar poucas arestas interpartições no processo de cópia. Como variáveis de restrições relaxadas participam de menos restrições dos subproblemas, o Subgradientes apresentou uma melhora no tempo de convergência e nos limites obtidos.

Para analisar a relação entre a quantidade de variáveis relaxadas e o valor do corte, é utilizada a proporção H = *nº de variáveis copiadas/corte*. Os dados com informações sobre os valores de *k*, quantidade de corte, restrições relaxadas e os valores de F_{bal} utilizados para obter a proporção H do particionamento desbalanceado, para todas as instâncias, são apesentados na Tabela 5.2. Os resultados obtidos com estes particionamentos são apresentados na Tabela 5.3.

Os valores de *k* da Tabela 5.3 apresentaram melhores limitantes para o problema. A otimalidade dos resultados foi comprovada para todas as instâncias e a versão desbalanceada da decomposição Lagrangeana obteve melhores resultados de tempo computacional. Além disto, foi possível obter novos valores para a função objetivo de três instâncias. Com os resultados apresentados, todas as instâncias geradas no trabalho de Yamamoto et al. (2002), e utilizadas por Mauri et al. (2010) tiveram suas soluções ótimas obtidas.

Tabela 5.2 – Resultados utilizando o particionamento apresentado por Mauri et al. (2010).

		Desbalanceado				Balanceado		
Instância	***k***	F_{bal}	**Corte**	**Variáveis Copiadas**	***H***	**Corte**	**Variáveis Copiadas**	***H***
1000_2	20	2	102	99	0.97	270	168	0.7
1000_4	26	1.7	161	142	0.88	448	246	0.54
1000_5	13	2	75	73	0.97	132	109	0.82
1000_7	20	1.7	135	105	0.77	176	159	0.9
1000_8	20	2	97	98	1.01	282	171	0.6
1000_9	20	2	123	127	1.03	201	162	0.8
1000_10	20	2	110	105	0.95	244	166	0.68
1000_12	16	3	65	73	1.12	195	148	0.7
1000_13	14	2	88	89	1.01	162	140	0.86
1000_14	10	3	44	43	0.97	89	77	0.86
1000_15	26	1.7	164	141	0.85	251	188	0.74
1000_16	25	2	194	181	0.93	312	234	0.75
1000_17	30	1.7	164	150	0.91	802	376	0.46
1000_18	25	2	170	154	0.9	277	212	0.76
1000_19	18	2	166	136	0.81	256	182	0.71
1000_21	24	2	131	119	0.9	377	234	0.62
1000_22	28	2	224	186	0.83	410	262	0.63
1000_23	26	2	176	167	0.94	397	273	0.68

Tabela 5.3 – Resultados com soluções ótimas obtidos com os particionamentos da Tabela 5.2.

	Desbalanceado					**Balanceado**			
Instância	***k***	**LI**	**LS**	***Gap***	**Tempo de execução**	**LI**	**LS**	***Gap***	**Tempo de execução**
1000_2	20	934	934	0	**462.78**	934	936.89	0.3	11750.23
1000_4	26	**934**	934.79	0.08	**805.10**	934	934.96	0.1	4260.88
1000_5	13	961	961.99	0.1	**1220.65**	961	961.22	0.02	6880.11
1000_7	20	929	929.98	0.1	**663.49**	929	929.97	0.1	3638.07
1000_8	20	940	940	0	**1561.10**	940	942.21	0.2	14419.89
1000_9	20	927	927.92	0.09	**4728.74**	925	959.52	3.73	8894.97
1000_10	20	944	944.89	0.09	**5483.18**	944	971.54	2.91	8861.16
1000_12	16	935	935	0	**196.13**	935	935.57	0.06	2660.74
1000_13	14	**955**	955	0	**136.88**	955	955.03	0.003	4623.14
1000_14	10	933	933	0	**4104.70**	933	934.87	0.2	10395.89
1000_15	26	934	934.98	0.1	**1246.76**	933	934.03	0.11	12216.92
1000_16	25	932	932.97	0.1	**3030.37**	931	959.58	3.06	9647.72
1000_17	30	937	937	0	**169.36**	937	940.45	0.36	14566.11
1000_18	25	946	946.97	0.1	**1132.63**	946	947.27	0.13	14509.09
1000_19	18	950	950	0	**244.11**	940	950.49	1.11	20842.81
1000_21	24	930	930	0	**174.97**	930	931.96	0.21	6595.36
1000_22	28	952	952.98	0.1	**5729.44**	952	953.57	0.16	18134.88
1000_23	26	934	934	0	**264.17**	934	935	0.1	8965.35
1000_24	12	**932**	932	0	**15172.38**	931	938.28	0.78	21354.97

5.4 Considerações finais

Este capítulo apresentou uma abordagem de Decomposição Lagrangeana desbalanceada para o Problema de Rotulação Cartográfica de Pontos. Para a abordagem inomogênea, foi implementado um algoritmo de particionamento multinível, que permite o desbalanceamento de partições, para que o corte de arestas interpartição seja menor, com o objetivo de obter menos restrições relaxadas na Decomposição Lagrangeana.

Através de testes computacionais, foi observado que a versão desbalanceada obteve melhores resultados de limitante, permitindo encontrar a solução ótima de todas as instâncias de testes, indicando que o desbalanceamento de partições melhora o valor de *gap* dos limites LI e LS, e o tempo de convergência destes limites, na execução do método de Subgradientes, por reduzir a quantidade de restrições relaxadas, mesmo aumentando o tamanho de alguns dos subproblemas. Observa-se porém que obter particionamento ótimo, não é o escopo deste trabalho, e portanto, é possível que resultados melhores de corte destas instâncias sejam obtidos.

Os bons resultados obtidos indicam que a Decomposição Lagrangeana desbalanceada pode ser utilizada para a futura resolução de instâncias maiores e mais complexas da abordagem de dispersão discreta para o PRCP.

6 CONCLUSÃO

Este trabalho de tese apresentou uma pesquisa em que o tema central é o Problema de Rotulação Cartográfica de Pontos. Como metodologia, o PRCP foi investigado como um problema de otimização combinatória, com a proposição de uma nova abordagem do problema e a adição de um novo critério de avaliação de soluções do PRCP. Além disso, como passo inicial para a utilização desta nova abordagem em instâncias maiores, foram apresentadas implementações de métodos que permitam a resolução do problema em larga escala.

Com relação à nova abordagem, foi proposta a modelagem do PRCP como um problema de Dispersão Discreta, em que distâncias de rótulos são utilizadas, ao invés de considerar apenas a estrutura de conflitos do grafo gerado pelo problema.

Com esta nova abordagem, tenta-se posicionar rótulos de maneira esparsa, com o intuito de obter uma melhor legibilidade das soluções, e indiretamente evitar conflitos. Para a avaliação dos resultados, foi considerado o valor da menor distância entre rótulos obtida como uma medida de dispersão.

Para a implementação, foram desenvolvidos três novos modelos de programação linear, que foram aplicados a instâncias obtidas da literatura, e comparados com uma formulação que obteve comprovadamente bons resultados, com relação aos critérios tradicionais, de proporção de rótulos livres em soluções do PRCP.

Os resultados mostraram que a abordagem dispersiva pode, mesmo que indiretamente, resolver bem problemas de conflitos de rótulos, e apresentar bons resultados de rótulos livres de conflitos, como os apresentados pelo MD2.

Como uma forma de utilizar ambos os critérios da medida de dispersão e a proporção de rótulos livres, a terceira formulação denominada MD3 tem em sua função objetivo a utilização do número de conflitos e das distâncias entre rótulos.

No entanto, a restrição (3.16) do MD3 pode gerar um alto custo computacional em instâncias com maior quantidade de conflitos e por isso, foi implementado um AGC para obter soluções de instâncias mais complexas. O AGC foi validado através da comparação dos seus resultados com os obtidos pelo CPLEX. Observa-se que as soluções do AGC foram melhores nos casos das instâncias maiores de 5046 pontos, indicando que a utilização de métodos evolutivos pode ser eficaz para a resolução de problemas de grande escala.

Além disso, para a futura resolução de instâncias maiores, como o conjunto completo de 13206 pontos do mapa da Suíça, foi apresentada uma melhoria da Decomposição Lagrangeana para o PMNRSC, ao utilizar o desbalanceamento dos subproblemas com o objetivo de obter uma menor quantidade de restrições relaxadas.

Através de testes computacionais, foi observado que a versão desbalanceada obteve melhores resultados de limitante, permitindo encontrar a solução ótima de todas as instâncias de testes, indicando que o desbalanceamento de partições melhora o tempo de convergência e o valor de *gap*, dos limites obtidos, ao utilizar o método de Subgradientes, por reduzir a quantidade de restrições relaxadas, mesmo aumentando o tamanho de alguns dos subproblemas. Observa-se, porém que obter o particionamento ótimo, não é o escopo deste trabalho e, portanto, é possível que resultados melhores de corte destas instâncias sejam obtidos. Os novos resultados apresentaram a solução ótima para todas as instâncias testadas.

6.1 Trabalhos futuros

Considerando os resultados obtidos através da investigação da abordagem dispersiva do PRCP, é sugerida a continuação da pesquisa envolvendo o tema. Possíveis extensões deste trabalho envolvem:

- Implementação de métodos para determinar automaticamente bons valores de k e de F_{bal}, que permitam obter bons resultados de corte de arestas interpartições e da proporção H.

- Utilizar a Decomposição Lagrangeana desbalanceada para a resolução do conjunto completo de 13206, que deu origem a diversas instâncias de teste utilizadas neste trabalho.

- Implementar a Decomposição Lagrangeana, ou outros métodos de decomposição, como a Dantzig-Wolfe, para os modelos da abordagem de dispersão.

- Utilizar computação distribuída para a resolução em paralelo dos subproblemas da Decomposição Lagrangeana desbalanceada.

- Para os casos em que não for possível obter boas soluções com métodos de decomposição, testar técnicas de otimização evolutiva como o Algoritmo Genético Construtivo ou de meta-heurísticas como o *Adaptive Large Neighborhood Search* (ALNS).

- Aplicar a modelagem dispersiva à variante contínua do PRCP.

- Aplicar a modelagem dispersiva aos problemas de rotulação de outras características cartográficas.

REFERÊNCIAS BIBLIOGRÁFICAS

AGARWAL, P. K.; KREVELD, M. V.; SURI, S. Label placement by maximum independent set in rectangles. **COMPUTATIONAL GEOMETRY: THEORY AND APPLICATIONS**, n. 11, p. 209-218, 1998.

ALVIM, A. C. F.; TAILLARD, E. D. **POPMUSIC for the point feature label placement.** In: In: METAHEURISTICS INTERNATIONAL CONFERENCE, 6., 2005, Viena, Austria. **Proceedings**…Viena, p. 39-44.

BUI, T. N.; JONES, C. Finding good approximate vertex and edge partitions is NP-Hard. **Information Processing Letters**, v. 42, p. 153-159, 1993.

CHRISTENSEN, J.; MARKS, J.; SHIEBER, S. An empirical study of algorithms for point-features label placement. **ACM TRANSACTIONS ON GRAPHICS**, v. 14, n. 3, p. 203–232, 1995.

CRAVO, G. L.; RIBEIRO, G. M.; LORENA, L. A. N. . A greedy randomized adaptive search procedure for the point-feature cartographic label placement. **Computers & Geosciences** , v. 34, n. 4, p. 373-386, 2008.

CURTIN, K. M.; CHURCH, R. L. A family of location models for multiple-type discrete dispersion. **Geographical Analysis**, v. 38, p. 248–270, 2006.

DIESTEL, R. **Graph Theory**. 2. ed. New York: Springer, 2000.

EDMONSON, S. et al. A general cartographic labeling algorithm. **Cartographica**, v. 33, n. 4, p. 12-23, 1996.

ERKUT, E.; NEUMAN, S. Comparison of four models for dispersing facilities.'. **INFOR**, v. 29, n. 2, p. 68-86, 1990.

ERKUT, E.; REVELLE, C.; ÜLKÜSAL, Y. Integer-friendly formulations for the r-Separation Problem. **European Journal of Operational Research**, v. 92, n. 2, p. 342-351, 1996.

FOGEL, D. B. **Evolutionary computation:** The fossil record. New York: IEEE Press, 1998.

GOMES, S. P.; RIBEIRO, G. M.; LORENA, L. A. N. Dispersion for the point feature cartographic label placement problem. **Expert Systems with Applications**, p. 5878-5883, 2013.

GOMES, S. P.; RIBEIRO, G. M.; LORENA, L. A. N. **Duas Novas Abordagens para o Problema de Rotulação Cartográfica de Pontos:** r-separação e p-

dispersão. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. Salvador, 2014.

GUIGNARD, M. Lagrangean relaxation. **TOP**, v. 11, n. 2, p. 151-228, 2003.

GUIGNARD, M.; KIM, S. Lagrangean decomposition: a model yielding stronger lagrangean bounds. **Mathematical Programming**, North-Holland, v. 39, p. 215-228, 1987.

HENDRICKSON, B.; LELAND, R. **A multilevel algorithm for partitioning graphs.** Supercomputing '95. New York: ACM Press. 1995.

HOLLAND, J. H. **Adaptation in natural and artificial systems.** Cambridge, Massachusetts, p. 11-147. 1975.

NTERNATIONAL BUSINESS MACHINE (IBM). **IBM ILOG CPLEX Optimizer 12.1**, 2009. Disponivel em: <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>. Acesso em: 5 Fevereiro 2015.

IMHOF, E. Positioning names on maps. **American Cartographer**, v. 2, n. 2, p. 128-144, 1975.

KARYPIS, G.; KUMAR, V. Multilevel k-way partitioning scheme for irregular graphs. **Journal of Parallel and Distributed Computing**, v. 48, n. 1, p. 96-129, 1998.

KERNIGHAN, B. W.; LIN, S. An Efficient Heuristic for Partitioning Graphs. **Bell Systems Technical Journal**, v. 49, p. 291-308, 1970.

KUBY, M. J. Programming models for facility dispersion: The p-dispersion and Maxisum dispersion problems. **Geographical Analysis**, v. 19, n. 4, p. 315-329, 1987.

LORENA, L. A. N.; FURTADO, J. C. Constructive genetic algorithm for clustering problems. **Evolutionary Computation**, v. 9, n. 3, p. 309-327, 2001.

MAURI, G. R.; RIBEIRO, G. M.; LORENA, L. A. N. A new mathematical model and a lagrangean decomposition for the point-feature cartographic label placement problem. **Computers & Operations Research** , v. 37, n. 12, p. 2164-2172, 2010.

MITCHELL, M. **An Introduction to Genetic Algorithms**. Cambridge: MIT Press, 1996.

MOON, I. D.; CHAUDHRY. An analysis of network location problems with distance constraints. **Management Science**, v. 30, n. 3, p. 290-307, 1984.

NARCISO, M. G.; LORENA, L. A. N. Lagrangean/surrogate relaxation for generalized assignment problems. **European Journal of Operational Research**, v. 114, n. 1, p. 165-177, 1999.

RIBEIRO, G. M. **Relaxação Lagrangeana com divisão em clusters para alguns problemas de otimização modelados em grafos de conflitos.** l 2007. 191 p. (INPE-15204-TDI/1304). Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2007. Disponível em: <http://urlib.net/sid.inpe.br/mtc-m17@80/2007/12.18.18.31.31>. Acesso em: 18 maio 2015.

RIBEIRO, G. M.; LORENA, L. A. N. Heuristics for cartographic label placement problems. **Computers & Geosciences**, v. 32, n. 6, p. 739-748, 2006.

RIBEIRO, G. M.; LORENA, L. A. N. Lagrangean relaxation with clusters for point-feature cartographic label placement problems. **Computers & Operations Research**, v. 35, n. 7, p. 2129-2140, 2008.

RIBEIRO, G. M.; MAURI, G. R.; LORENA, L. A. N. A lagrangean decomposition for the maximum independent set problem applied to map labeling.. **Operational Research**, v. 11, n. 3, p. 229-243, 2011.

RYLOV, M. A.; REIMER, A. W. A comprehensive multi-criteria model for high cartographic quality point-feature label placement. **Cartographica**, v. 49, n. 1, p. 52-68, 2014.

SACHDEVA, S. **Development of a branch and price approach involving vertex cloning to solve the maximum weghted independent set problem.** 2004. Thesis (Master Science in Industrial Engineering) - A&M University, 2004.

STRIJK, T.; VERWEIJ, B.; AARDAL, K. Algorithms For Maximum Independent Set Applied To Map Labeling. **Universiteit Utrecht:** Institute of Information and Computing Sciences, 2000. Disponivel em: <ftp://ftp.cs.uu.nl/pub/ruu/cstechreps/cs-2000/2000-22.ps.gz>.

VAN DIJK, S. et al. Towards an evaluation of quality for names placement methods. **International journal of geographical information science**, v. 16, n. 7, p. 641-661, 2002.

VAN KREVELD, M. J.; STRIJK, T. W.; WOLFF, A. Point labeling wiht sliding labels. **Computational Geometry: Theory and applications**, v. 13, n. 1, p. 21-47, 1999.

VERNER, O. V. Placing text labels on maps and diagrams using genetic algorithms with masking. **Informs Journal on Computing**, p. 266–275, 1997.

VERWEIJ, A. M.; AARDAL, K. I. An optimization algorithm for maximum independent set with aplications in map labelling. **Lecture Notes in Computer Science**. v. 1643, p. 426-437,1999.

WAGNER, F. et al. Three rules suffice for good label placement. **Algorithmica** , v. 30, p. 334-349, 2001.

YAMAMOTO, M.; CAMARA, G.; LORENA, L. A. N. Tabu search heuristic for point-feature cartographic label placement. **GEOINFORMATICA**, v. 6, n. 1, p. 77–90, 2002.

YAMAMOTO, M.; LORENA, L. A. N. A constructive genetic approach to point-feature cartographic label placement. In: **METAHEURISTICS INTERNATIONAL CONFERENCE**, 35., 2003, Kyoto. **Proceedings...** 2003. On-line. (INPE-11778-PRE/7137). Disponível em:<http://urlib.net/sid.inpe.br/marciana/2004/03.15.15.33>. Acesso em: 18 maio 2015, p. 285-300, 2005.

ZORASTER, S. The solution of large 0-1 integer programming problems encountered in automated cartography. **OPERATIONS RESEARCH**, v. 38, n. 5, p. 752-759, 1990.