



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-15204-TDI/1304

**RELAXAÇÃO LAGRANGEANA COM DIVISÃO EM CLUSTERS
PARA ALGUNS PROBLEMAS DE OTIMIZAÇÃO MODELADOS
EM GRAFOS DE CONFLITOS**

Glaydston Mattos Ribeiro

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Luiz Antonio Nogueira Lorena, aprovada em 28 de novembro de 2007.

INPE
São José dos Campos
2008

Publicado por:

esta página é responsabilidade do SID

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) 3945-6911

Fax: (012) 3945-6919

E-mail: pubtc@sid.inpe.br

**Solicita-se intercâmbio
We ask for exchange**

Publicação Externa – É permitida sua reprodução para interessados.



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-15204-TDI/1304

**RELAXAÇÃO LAGRANGEANA COM DIVISÃO EM CLUSTERS
PARA ALGUNS PROBLEMAS DE OTIMIZAÇÃO MODELADOS
EM GRAFOS DE CONFLITOS**

Glaydston Mattos Ribeiro

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Luiz Antonio Nogueira Lorena, aprovada em 28 de novembro de 2007.

INPE
São José dos Campos
2008

Dados Internacionais de Catalogação na Publicação

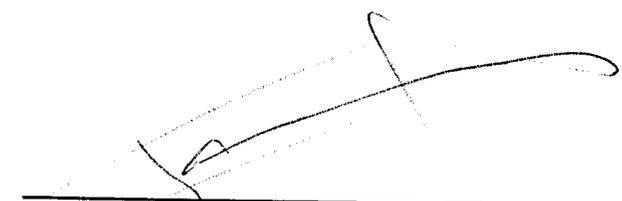
R354r Ribeiro, Glaydston Mattos
Relaxação Lagrangeana com divisão em clusters para alguns
problemas de otimização modelados em grafos de conflitos /
Glaydston Mattos Ribeiro. – São José dos a Campos: INPE, 2007.
191p. ; (INPE-15204-TDI/1304)

1.Relaxação Lagrangeana. 2.Relaxação Lagrangeana com
clusters. 3.Algoritmo de geração de colunas.4.Teoria de cortes.Título.

CDU 681.5.015.23

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

Dr. Horacio Hideki Yanasse



Presidente / INPE / SJC Campos - SP

Dr. Luiz Antonio Nogueira Lorena



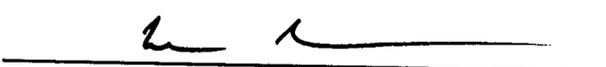
Orientador(a) / INPE / SJC Campos - SP

Dr. Edson Luiz Franca Senne



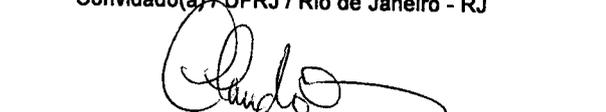
Convidado(a) / UNESP/GUARA / Guaratinguetá - SP

Dr. Nelson Maculan Filho

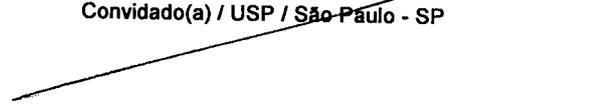


Convidado(a) / UFRJ / Rio de Janeiro - RJ

Dr. Cláudio Barbieri da Cunha



Convidado(a) / USP / São Paulo - SP



Aluno (a): Glaydston Mattos Ribeiro

São José dos Campos, 28 de Novembro de 2007

Dedico este trabalho a meus pais, meus irmãos e a minha esposa.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus por ter me dado força nos momentos complicados.

Agradeço imensamente ao meu orientador, Prof. Luiz Antonio Nogueira Lorena, pelo apoio, conhecimento transmitido e orientação.

Gostaria de deixar aqui o meu agradecimento especial ao Instituto Nacional de Pesquisas Especiais (INPE) e ao Laboratório Associado de Matemática e Computação Aplicada (LAC) pela oportunidade. E aproveitar para agradecer também aos professores do LAC pela convivência e ensinamentos transmitidos, em especial aos Profs. José Demisio Simões da Silva, José Carlos Becceneri, Horacio Hideki Yanasse e Edson Luiz França Senne.

Agradeço a todos que, de maneira direta ou indireta, ajudaram na elaboração deste trabalho, de modo especial:

A Universidade de Lisboa e o Centro de Investigação Operacional (CIO) pelo apoio durante meu período de estágio.

Os professores do CIO, em especial, o Prof. Miguel Fragoso Constantino pelo apoio e orientação.

Os membros da banca pela disposição e análise deste documento.

O Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e o Programa ADONET pelo auxílio financeiro.

A Universidade Federal do Espírito Santo (UFES), em especial, aos colegas do Centro Universitário Norte do Espírito Santo (CEUNES) que muito me ajudaram.

A minha esposa Graziela, pelo amor, carinho, apoio, compreensão, e principalmente por estar do meu lado.

O meu Pai, Francisco Glaucio, e minha mãe, Gracinha, que, mesmo não estando do meu lado, sempre me incentivaram. A minha mãe Marly que, mesmo de longe, me deu forças e rezou por mim.

Os meus irmãos, Ingrid, Thiago, Marcos, Fernanda e Aline, que entenderam minha ausência e sempre me apoiaram.

A Profa. Maria Inês Faé pelo incentivo e apoio desde os tempos da Iniciação Científica na UFES.

Os amigos Luizão (meu “irmão” mais velho), Almerinda, família e “agregados”, pelos bons tempos em São José dos Campos (churrascos, festas, pernoites, refeições, caronas,...).

Todos os amigos, em especial, Geraldo, Corrêa, Antônio, Fabrício, Agnaldo, Saul e Sérgio, pelo incentivo e força.

RESUMO

Muitos problemas de otimização combinatória podem ser modelados por meio de um grafo especial denominado grafo de conflitos. Quando estes grafos apresentam-se esparsos, bem adaptados para uma fase de particionamento, ou seja, quando apresentam agrupamentos de vértices bem definidos (*clusters*), as arestas que conectam os *clusters* podem ser relaxadas no sentido lagrangeano, e o problema relaxado pode ser decomposto e resolvido. Essa é a idéia da relaxação lagrangeana com clusters (LagClus) proposta neste trabalho. A vantagem desta relaxação reside no fato de que os subproblemas gerados apresentam estruturas semelhantes às do problema original, conseqüentemente, os limitantes duais obtidos são melhores que os de uma relaxação lagrangeana tradicional sobre todas as arestas do grafo de conflitos. A LagClus foi aplicada com êxito a Problemas como o da Rotulação Cartográfica de Pontos (PRCP), do Carregamento de Paletes do Produtor (PCPP), da Estivagem de Unidades de Celulose (PEUC) e da Programação Diária de Imagens de um Satélite de Observação (PPDISO). Por outro lado, a decomposição do grafo de conflitos permite com que o problema seja modelado segundo a decomposição Dantzig-Wolfe, em que o Problema Mestre Restrito (PMR) é formado pelas restrições de acoplamento (restrições de conexão dos *clusters*) e o subproblema gerador de colunas, pelos *clusters* definidos no particionamento. Resultados computacionais comprovam a equivalência entre a LagClus e essa decomposição. Seguindo a linha dos modelos matemáticos, este trabalho ainda apresenta novas formulações matemáticas para o PRCP, PEUC e PPDISO que foram baseadas na teoria de cortes. Resultados computacionais, principalmente para o PPDISO, mostram que a inserção de cortes adequados leva a formulações mais fortes que em um tempo computacional reduzido, permitem encontrar a solução ótima de várias instâncias presentes na literatura.¹

¹Este trabalho conta com apoio financeiro do CNPq e Programa ADONET-*An European Network in Algorithmic Discrete Optimization*, contrato MRTN-CT-2003-504438

LAGRANGEAN RELAXATION WITH CLUSTERS FOR SOME OPTIMIZATION PROBLEMS MODELED BY CONFLICT GRAPHS

ABSTRACT

Several combinatorial optimization problems can be modeled by a special graph denoted conflict graph. When these graphs are sparse well-adapted for a previous clustering phase, i.e, when they have clusters of vertices, the edges inter clusters can be relaxed in a lagrangean fashion, and the relaxed problem can be decomposed into sub problems and solved. This is the idea of the lagrangean relaxation with clusters (LagClus). The sub problems have similar structure of the original problem. Therefore the dual bounds are better than the traditional lagrangean relaxation over all edges of the conflict graph. This is the main advantage of the LagClus. We applied the LagClus over different problems present in the literature: Point-Feature Cartographic Label Placement Problem (PFCLP), Manufacturer's Pallet Loading Problem (MPLP), Woodpulp Stowage Problem (WSP) and Daily Photograph Scheduling Problem of an Earth Observation Satellite (DPSPEO). The decomposition of the conflict graph allows us to model the problem using the Dantzig-Wolfe decomposition, where the Restricted Master Problem (RMP) is defined by coupling constraints (all edges connecting the clusters) and the sub problem by the clusters obtained in the partitioning phase. Computational results show the equivalence between LagClus and this decomposition. This work also presents new mathematical formulations for three problems: PFCLP, WSP and DPSPEO. All these formulation are based on cutting theory. Computational tests, mainly for DPSPEO, show that when we insert interesting cuts generating new formulations, the optimal solutions can be found very quickly. ²

²This work was supported by CNPq and ADONET Program-An European Network in Algorithmic Discrete Optimization, contract MRTN-CT-2003-504438

SUMÁRIO

Pág.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SÍMBOLOS

LISTA DE SIGLAS E ABREVIATURAS

CAPÍTULO 1	INTRODUÇÃO	29
1.1	Representação em Grafos	30
1.2	Motivação	30
1.3	Objetivos e Contribuições	31
1.4	Organização da Tese	32
CAPÍTULO 2	FUNDAMENTAÇÃO TEÓRICA	33
2.1	Conceitos Sobre Relaxação	33
2.2	Relaxação Lagrangeana	35
2.2.1	Forma Geral e Dual Lagrangeano	35
2.2.2	Algoritmo de Subgradientes	37
2.3	Decomposição Dantzig-Wolfe	38
2.3.1	O Método de Geração de Colunas na Decomposição Dantzig-Wolfe	40
2.3.2	Algoritmo Básico de Geração de Colunas	41
2.4	Planos de Corte	42
2.4.1	Poliedros	43
2.4.2	Formulação “Ideal”	44
2.4.3	Algumas Desigualdades Válidas	45
2.5	Representação em Grafos	47
2.5.1	Grafo de Conflitos	47
2.5.2	Desigualdades Válidas para o PMCV	49
2.6	Particionamento de Grafos	51
2.6.1	Algoritmos Seqüenciais	52
2.6.2	Algoritmos Globais	53
2.6.3	Algoritmos Paralelos	53
2.6.4	Algoritmos de Multi-Níveis	54
2.6.5	Considerações Finais	57

CAPÍTULO 3	RELAXAÇÃO LAGRANGEANA COM CLUSTERS	59
3.1	A LagClus e Grafos de Conflitos	59
3.1.1	Principais Passos da LagClus	61
3.1.2	LagClus e o PMCIV	62
3.1.3	Aplicando a LagClus sobre uma Instância do PMCIV	63
3.1.4	Aspectos da LagClus que devem ser Observados	68
3.2	Algoritmo de Geração de Colunas	68
3.2.1	Aplicando o AGC sobre uma Instância do PMCIV	70
3.2.2	Aspectos que devem ser Observados no AGC que usa a Idéia dos <i>Clusters</i>	72
3.3	Considerações Finais do Capítulo	74
CAPÍTULO 4	ROTULAÇÃO CARTOGRÁFICA DE PONTOS . . .	77
4.1	Padronização Cartográfica e o Grafo de Conflitos do PRCP	78
4.2	Revisão Bibliográfica para o PRCP	79
4.3	Modelagem Matemática Baseada em Arestas - PMNC ^a	83
4.4	Modelagem Matemática Baseada em Pontos - PMNC ^p	84
4.5	Relaxações Lagrangeanas Consideradas para o PMNC ^a	86
4.5.1	Relaxação Lagrangeana Tradicional e LagSur	86
4.5.2	LagClus para o PMNC ^a	89
4.6	Relaxações Lagrangeanas Consideradas para o PMNC ^p	92
4.6.1	LagClus para o Modelo PMNC ^p	92
4.6.2	Modelo Matemático da LagClus para o PMNC ^p	93
4.7	Algoritmo de Geração de Colunas para o PMNC	94
4.8	Resultados Computacionais	96
4.8.1	Resultados Computacionais Sem a Redução do Grafo de Conflitos	96
4.8.2	Resultados Computacionais Com a Redução do Grafo de Conflitos	103
4.8.3	Resultados Computacionais do AGC	105
4.8.4	Comparação com a Literatura	108
4.9	Considerações Finais	108
CAPÍTULO 5	CARREGAMENTO DE PALETES DO PRODUTOR E ESTIVAGEM DE UNIDADES DE CELULOSE	111
5.1	Introdução sobre o PCPP	111
5.1.1	Revisão Bibliográfica para o PCPP	112
5.1.2	Formulação do PPCP	115
5.1.3	LagClus para o PCPP	120
5.1.4	AGC para o PCPP usando a Idéia dos <i>Clusters</i>	123

5.1.5	Resultados Computacionais	125
5.1.6	Resultados da LagClus	127
5.2	Resultados do AGC	130
5.3	Introdução sobre o PEUC	131
5.3.1	Formulação do PEUC	134
5.3.2	LagClus para o PEUC	134
5.3.3	Experimentos Computacionais	135
5.4	Considerações Finais	138
CAPÍTULO 6 PROGRAMAÇÃO DIÁRIA DE UM SATÉLITE DE OBSERVAÇÃO		141
6.1	Revisão Bibliográfica para o PPDI-S5	142
6.2	Formulação Matemática para o PPDI-S5	143
6.3	Relaxações Lagrangeanas para o PPDI-S5	145
6.3.1	LagClus para o PPDI-S5	145
6.3.2	Relaxação Lagrangeana Sobre as Restrições Ternárias	148
6.4	Novas Formulações para o PPDI-S5	148
6.4.1	Formulação Usando Cliques Maximais Envolvendo um Vértice e Cobertura de Vértices	149
6.4.2	Formulação Usando Cliques Maximais (PPDI-S5_M ₂)	153
6.4.3	Formulação Usando Cliques Maximais e Situação Par-Binário (PPDI-S5_M ₃)	154
6.4.4	Formulação Usando Cliques Maximais e Ternárias com Cliques (PPDI-S5_M ₄)	156
6.4.5	Formulação Usando Cliques Maximais e “Cliques Maximais Ternárias” a partir de Restrições Binárias (PPDI-S5_M ₅)	158
6.4.6	Formulação Usando Cliques Maximais e “Cliques Maximais Ternárias” a partir de “Grafos Ternários” (PPDI-S5_M ₆)	160
6.5	Resultados Computacionais	163
6.5.1	Resultados das Relaxações Lagrangeanas	163
6.5.2	Resultados com os Novos Modelos para o PPDI-S5	166
6.5.3	Comparação com a Literatura	172
6.6	Considerações Finais	172
CAPÍTULO 7 CONCLUSÕES E TRABALHOS FUTUROS		173
7.1	Resumos das Contribuições	174
7.1.1	Modelos Matemáticos	174
7.1.2	Relaxação Lagrangeana com <i>Clusters</i> (LagClus)	175

7.1.3	Algoritmo de Geração de Colunas	176
7.2	Trabalhos Futuros	176
	REFERÊNCIAS BIBLIOGRÁFICAS	179

LISTA DE FIGURAS

	Pág.
2.1 Algoritmo de Subgradientes segundo Parker e Rardin (1988).	38
2.2 Algoritmo básico de Geração de Colunas.	42
2.3 Exemplo de um poliedro no \Re^n	43
2.4 Poliedros das formulações F^1 , F^2 e F^3	44
2.5 Exemplo de um grafo $G = (V, A)$	47
2.6 PMCIV. (a) Grafo de conflitos, (b) formulação matemática e (c) solução ótima do problema.	49
2.7 Exemplo de <i>1-roda simples</i> (Cheng e Cunningham, 1997).	51
2.8 Principais passos dos algoritmos de multi-níveis para o PPG descritos por Karypis (2002)	55
2.9 Formato de arquivos no METIS para vários tipos de grafos.	56
3.1 Exemplo de um grafo de conflitos com regiões bem definidas: (a) grafo de conflitos, (b) arestas de conexão e (c) <i>clusters</i> independentes.	60
3.2 Um exemplo prático da aplicação da LagClus sobre uma instância do PMCIV.	64
3.3 Adaptação do algoritmo de subgradientes para contemplar a LagClus.	67
3.4 AGC alterado para contemplar o limitante da LagClus.	73
4.1 Um exemplo de um mapa com rótulos sobrepostos (veja setas).	77
4.2 Padronização cartográfica proposta por Christensen et al. (1995).	78
4.3 Exemplo de um grafo de conflitos para o PRCP: (a) Posições candidatas, (b) grafo de conflitos e (c) solução ótima.	79
4.4 Comparação entre o PMNRSC e PMNC.	81

4.5	Regras para reduzir o grafo de conflitos propostas por Wagner et al. (2001)	82
4.6	Heurística lagrangeana de melhoria (HM) aplicada no algoritmo de subgradientes	87
4.7	Heurística de Construção (HC) aplicada no algoritmo de subgradientes.	89
4.8	Particionamento do grafo de conflitos. (a) Problema, (b) grafo de conflitos e os <i>clusters</i> , (c) arestas entre <i>clusters</i> , e (d) subproblemas gerados.	89
4.9	Exemplo de aplicação da LagClus para a formulação baseada em pontos.	92
4.10	Um exemplo para ilustrar as variáveis e matrizes presentes na LagClus para o modelo PMNC ^p .	95
4.11	Diagrama dos passos da AGC para o PMNC ^p .	105
4.12	Algoritmo para gerar o conjunto inicial de colunas.	106
5.1	Tipos de PCP: (a) Problema do Carregamento de Paletes do Produtor e (b) Problema do Carregamento de Paletes do Distribuidor (Adaptado de Morabito e Morales (1998)).	112
5.2	Heurística de Blocos. (a) 1-Bloco, (b) 2-Blocos, (c) 3-Blocos, (d) 4-Blocos e (e) 5-Blocos. (Adaptado de Martins (2003)).	113
5.3	Pseudocódigo da Heurística de 5 Blocos.	114
5.4	Posição (r, s) não permitida em função da colocação de uma caixa na posição (p, q) com orientação i e (b) possíveis orientações para uma caixa na modelagem do PCPP.	116
5.5	Solução ótima para o problema $(L, W) = (5, 4)$ e $(l, w) = (3, 2)$.	119
5.6	Comparação entre a formulação de Beasley (1985) e a do PMCIV. (a) Formulação de Beasley (1985), (b) grafo de conflitos e (c) PMCIV.	120
5.7	Exemplo de como decompor uma clique relaxada nos <i>clusters</i> .	121
5.8	Heurística de Verificação e Melhoria (HVM) utilizada no algoritmo de subgradientes para o PCPP.	123
5.9	Diagrama dos passos da AGC para o PCPP.	125

5.10	Comportamento do problema mestre restrito e da LagClus obtida a partir do multiplicadores duais do PMR.	131
5.11	Estivagem de unidades de celulose em porões de navios	132
5.12	Unidade de celulose (a), disposição dos porões e (c) exemplo de uma estivagem	133
5.13	Grafo de conflitos gerado para a instância real P2: 22.870 vértices e 4.792.819 arestas	135
5.14	Planos de estivagem praticados nos portos e os novos gerados pelo LagClus	138
6.1	Grafo de conflitos com restrições binárias e ternárias.	146
6.2	Clique maximal envolvendo o vértice x_1	149
6.3	Exemplo de aplicação da redução de Murray e Church (1997a).	150
6.4	Processo de geração das restrições de Murray e Church (1997a).	152
6.5	Desigualdade válida para o PPDI-S5.	154
6.6	Inequações válidas para o PPDI-S5 usando restrições ternárias e cliques.	156
6.7	Inequação válida para o PPDI-S5 usando as “cliques maximais ternárias” a partir das restrições binárias.	159
6.8	Algoritmo para gerar as “cliques maximais ternárias” a partir de restrições binárias.	159
6.9	Inequação válida para o PPDI-S5 usando “cliques maximais ternárias” obtidas a partir de “grafos ternários”.	161
6.10	Algoritmo para gerar “cliques maximais ternárias” a partir de “grafos ternários”.	162

LISTA DE TABELAS

	Pág.
4.1 Número médio de restrições geradas.	85
4.2 Resultados médios do CPLEX 7.5 com o modelo PMNC ^a	97
4.3 Resultados médios do CPLEX 7.5 com o modelo PMNC ^p	97
4.4 Resultados médios para o limitante inferior das relaxações lagrangeana e LagSur sobre as restrições 4.6.	99
4.5 Resultados médios para o limitante superior das relaxações lagrangeana e LagSur sobre as restrições 4.6.	99
4.6 Iterações e tempos computacionais das relaxações lagrangeana e LagSur sobre as restrições 4.6.	100
4.7 Iterações e tempos computacionais da LagSur sobre as restrições 4.5.	100
4.8 Informações sobre o particionamento realizado para a LagClus com o modelo PMNC ^a	101
4.9 Resultados médios obtidos com a LagClus sobre o PMNC ^a	101
4.10 Informações sobre o particionamento realizado para a LagClus com o modelo PMNC ^p	102
4.11 Resultados médios obtidos com a LagClus sobre o PMNC ^a	102
4.12 Resultados médios da LagClus sobre o PMNC ^p variando o número de <i>clusters</i> para os problemas com 1000 pontos.	102
4.13 Informações sobre o processo de redução do grafo de conflitos.	103
4.14 Resultados médios do CPLEX 10 com o modelo PMNC ^p usando grafos reduzidos.	103
4.15 Resultados médios da LagClus com o modelo PMNC ^p sobre os grafos reduzidos.	104

4.16	Resultados médios da LagClus com o modelo $PMNC^p$ sobre os grafos reduzidos.	107
4.17	Comparação com outros algoritmos.	108
5.1	Resultados computacionais para 10 instâncias propostas por Letchford e Amaral (2001), consideradas difíceis para uma relaxação lagrangeana.	128
5.2	Resultados computacionais para as 10 instâncias aleatoriamente escolhidas da COVER II (Dowland, 1987b).	128
5.3	Resultados computacionais para as 10 instâncias aleatoriamente escolhidas da COVER III (Alvarez-Valdes et al., 2005a), sobre todas aquelas que não apresentam solução ótima conhecida.	129
5.4	Resultados computacionais da LagClus para a instância L7 variando o número de <i>clusters</i>	129
5.5	Resultados obtidos com a Geração de Colunas.	130
5.6	Instâncias consideradas para o PEUC.	136
5.7	Resultados da LagClus para as instâncias do PEUC.	136
5.8	Análise do embarque de unidades de celulose.	137
6.1	Resultados das relaxações de PL e da LagClus.	164
6.2	Resultados da LagClus-Peso.	165
6.3	Resultados com $L_\lambda PPDIS-S5$	166
6.4	Número de restrições geradas por cada modelo.	167
6.5	Resultados da relaxação de PL.	168
6.6	Resultados computacionais para um 0,01% de <i>gap</i>	169
6.7	Resultados computacionais com o CPLEX trabalhando no máximo 7200s.	169
6.8	Resultados computacionais do CPLEX até resolver o problema ou parar por falta de memória.	170

6.9 CPLEX sem a sua geração própria de cortes.	171
6.10 Comparação com a literatura.	171

LISTA DE SÍMBOLOS

A	– Conjunto de arestas de um grafo
A^p	– Matriz binária de coeficientes das variáveis do <i>cluster</i> p pertencentes às arestas entre <i>clusters</i>
B^n	– Vetor binário de dimensão n
CL_k	– Conjunto de vértices que compõem a clique maximal k
$CP_{i,j}$	– Conjunto de todos os pontos que possuem posições candidatas conflitantes com a posição $x_{i,j}$
DL	– Problema dual lagrangeano
D^p	– Matriz de coeficientes das restrições do <i>cluster</i> p
D_t	– Dual da relaxação lagrangeana/ <i>surrogate</i>
G	– Representação de um grafo
K	– Conjunto de todas as cliques maximais
$L_\lambda P$	– Relaxação lagrangeana do problema P
$LG_\lambda P$	– Relaxação lagrangeana com <i>clusters</i> do problema P
$L_t SP^\lambda$	– Relaxação lagrangeana/ <i>surrogate</i> do problema P
N	– Número de pontos a serem rotulados
\tilde{N}_i	– Conjunto de vértices que possuem uma restrição de adjacência com o vértice i
P	– Problema de otimização
P_i	– Número de posições candidatas de um ponto i
PI	– Problema de otimização com variáveis inteiras
P_{DW}	– Decomposição Dantzig-Wolfe do problema P
\bar{P}	– Número de partições de um grafo
\Re^n	– Vetor de números reais de dimensão n
\Re_+^n	– Vetor de número reais não negativos de dimensão n
$S_{i,j}$	– Conjunto de pares de posições candidatas conflitantes com a posição candidata $x_{i,j}$
V	– Conjunto de vértices de um grafo
V'	– Subconjunto de vértices do conjunto V
$w_{i,j}$	– Custo associado à posição candidata $x_{i,j}$
Z^+	– Conjunto dos números inteiros não negativos
Z_+^n	– Vetor de números inteiros não negativos de comprimento n
β_p	– Variável dual associada à p -ésima restrição
$\ x\ $	– Norma do vetor x

LISTA DE SIGLAS E ABREVIATURAS

3D-BPP	–	<i>Three-Dimensional Bin Packing Problem</i>
ACS	–	<i>Ant Colony System</i>
AGC	–	Algoritmo de Geração de Colunas
B&P	–	<i>Branch-and-Price</i>
D–W	–	Decomposição Dantzig-Wolfe
GRASP	–	<i>Greedy Randomized Adaptive Search Procedure</i>
LagClus	–	Relaxação Lagrangeana com <i>Clusters</i>
LagSur	–	Relaxação Lagrangeana/ <i>Surrogate</i>
OC	–	Otimização Combinatória
KL	–	Algoritmo de Kernighan e Lin (1970)
PCP	–	Problema do Carregamento de Paletes
PCPP	–	Problema do Carregamento de Paletes do Produtor
PCPD	–	Problema do Carregamento de Paletes do Distribuidor
PCV	–	Problema do Caixeiro Viajante
PEUC	–	Problema da Estivagem de Unidades de Celulose
PM	–	Problema Mestre
PMCIIV	–	Problema do Máximo Conjunto Independente de Vértices
PMCIIVP	–	Problema do Máximo Conjunto Independente de Vértices com Pesos
PMR	–	Problema Mestre Restrito
POPMUSIC	–	<i>Partial Optimization Metaheuristic under Special Intensification Conditions</i>
PPDISO	–	Problema da Programação Diária de Imagens de um Satélite de Observação
PPDI-S5	–	Problema da Programação Diária de Imagens do Satélite SPOT 5
PPG	–	Problema de Particionamento de Grafos
PRCP	–	Problema da Rotulação Cartográfica de Pontos
PMNRSC	–	Problema do Máximo Número de Rótulos Sem Conflitos
PMNC	–	Problema da Minimização do Número de Conflitos
RSF	–	<i>Recursive-Smallest-First</i>

CAPÍTULO 1

INTRODUÇÃO

Muitos problemas empresariais requerem soluções baseadas em decisões discretas como, por exemplo, construir ou não um novo centro de distribuição, visitar o porto A ao invés do B ou realizar a seqüência 1 ao invés da 2. Entretanto, essas decisões normalmente estão associadas a algum tipo de avaliação que na maioria das vezes é monetária, por exemplo, “Dado um conjunto de 10 locais candidatos que podem receber um centro de distribuição, qual deles é o que apresenta o menor custo de instalação para um dado nível de serviço?”.

Assim, dependendo do número de soluções discretas, um tomador de decisões pode enumerá-las manualmente e definir aquela que melhor atende aos objetivos da empresa. Porém, quando o número de decisões discretas é muito grande, o número de combinações pode crescer rapidamente e talvez inviabilizar um processo manual de solução (Parker e Rardin, 1988).

O estudo dessas combinações em um problema de otimização discreta é o foco da Otimização Combinatória (OC) que, na sua forma geral, tem como objetivo maximizar (ou minimizar) uma função sujeita a restrições. Na literatura existem vários problemas de OC amplamente estudados como o problema do caixeiro viajante, o da mochila e de localização de facilidades, que ainda apresentam-se desafiadores.

Devido às dificuldades intrínsecas encontradas em muitos problemas de OC, muitas técnicas de solução vêm sendo aprimoradas ou desenvolvidas, englobando desde modelos matemáticos mais ajustados, passando por relaxações, até métodos de decomposição bem fundamentados teoricamente (Wolsey, 1998).

No campo dos modelos matemáticos, técnicas vêm sendo estudadas para criar ou fortalecer as formulações. Fortalecer aqui se refere ao estudo de novas restrições que eliminam certas soluções não factíveis como as desigualdades de Chvátal-Gomory (Veja Wolsey (1998)). Com relação às relaxações, as mais utilizadas são a relaxação de programação linear, Lagrangeana, Surrogate e Lagrangeana/*Surrogate* (Narciso e Lorena, 1999). Dentre estas, destaca-se a relaxação Lagrangeana por ter inúmeras aplicações na literatura.

As soluções das relaxações fornecem limitantes que podem ser utilizados para verificar a qualidade de uma solução factível. Esses limitantes permitem garantir, em alguns casos, se uma solução factível é ou não a melhor solução para o problema em questão (solução

ótima).

Sob o foco de decomposição, vários métodos vêm sendo estudados como, por exemplo, Algoritmos de Geração de Colunas (AGC), Algoritmos de *Branch-and-Cut* e Algoritmos de *Branch-and-Price*. Esses algoritmos aproveitam alguma informação do problema original para decompô-lo em problemas independentes.

1.1 Representação em Grafos

Decisões discretas, por exemplo, do tipo “Sim ou Não”, podem ser representadas por variáveis binárias que assumem o valor 1 indicando o estado “Sim” e 0, caso contrário. Essa representação permite com que problemas de OC eventualmente sejam definidos através de grafos, facilitando o entendimento e o desenvolvimento de técnicas de solução.

Nesses grafos cada aresta representa uma relação lógica entre variáveis (vértices do grafo). Dependendo da aplicação, arestas podem ou não ter direção, pode ser permitido que arestas liguem um vértice a ele próprio e vértices e/ou arestas podem ter um valor numérico associado (peso). Uma aresta entre dois vértices i e j pode indicar, por exemplo, que as respectivas variáveis de decisão não podem aparecer simultaneamente na solução ótima do problema e nesse caso, tem-se um grafo de conflitos.

Ao representar um problema de OC por meio de um grafo, agrupamentos (*clusters*) de vértices podem surgir explicitamente. Esses *clusters* indicam que as suas variáveis estão mais fortemente relacionadas entre si.

Sendo assim, se as arestas que mantêm esses *clusters* conectados forem removidas, cada um deles pode ser considerado independente e resolvido por algum aplicativo comercial. Essa estratégia, baseada no princípio de “dividir para conquistar”, não garante resolver o problema original, mas pode produzir um limitante interessante, pois se trata de uma relaxação.

1.2 Motivação

Atualmente, existem vários aplicativos comerciais que utilizam as melhores técnicas presentes na literatura para a resolução de problemas de OC. Esses aplicativos resolvem problemas de pequeno a grande porte em um tempo computacional aceitável, porém, mesmo com o avanço tecnológico obtido nos últimos anos, muitos problemas práticos ainda não são otimamente resolvidos.

Alguns problemas práticos possuem milhares de variáveis e restrições, o que inviabiliza

muitos métodos de solução. Por outro lado, como visto na Seção 1.1, quando se é considerado o grafo induzido pelo problema, pode-se pensar em usar a estratégia de “dividir para conquistar”.

Essa estratégia pode ser explorada em uma relaxação ou até mesmo em um método de decomposição. Além disso, a representação em grafo permite estudar técnicas de fortalecimento como as cliques usadas em problemas de empacotamento de nós (*Node Packing*) (Nemhauser e Sigismondi, 1992), para obter formulações que potencialmente fornecem limitantes melhores.

A carência de métodos eficientes para a solução desses problemas desafiadores e a própria falta de estratégias para gerar bons limitantes que permitam avaliar uma solução factível, constituem a principal motivação deste trabalho.

1.3 Objetivos e Contribuições

Esta tese tem como objetivo apresentar contribuições relacionadas a problemas de otimização combinatória modelados em grafos de conflitos. Essas contribuições estão divididas em três grandes áreas:

- 1^a Modelos Matemáticos. São apresentadas algumas novas formulações para o Problema da Rotulação Cartográfica de Pontos (PRCP) (Hirsh, 1982; Zoraster, 1990), para o Problema da Estivagem de Unidades de Celulose (PEUC) (Ribeiro e Lorena, 2006d) e para o Problema da Programação Diária de Imagens de um Satélite de Observação (PPDISO) (Bensana et al., 1999).
- 2^a Relaxação Lagrangeana com *Clusters*. É apresentada uma nova relaxação Lagrangeana, denominada relaxação Lagrangeana com *Clusters* (LagClus), que é baseada na decomposição do problema original (grafo) em *clusters* (subproblemas). A LagClus foi aplicada com sucesso em alguns problemas de OC sendo eles: PRCP, o Problema de Carregamento de Paletes de Produtor (PCPP) (Hodgson, 1982; Dowsland, 1987b), PEUC e PPDISO.
- 3^a Geração de Colunas. Considerando a LagClus e sua estratégia de decomposição, esta tese ainda apresenta um Algoritmo de Geração de Colunas (AGC) que utiliza os *clusters* (subproblemas) para gerar novas colunas para um Problema Mestre Restrito (PMR). O AGC foi também aplicado com sucesso no PRCP e PCPP, se mostrando mais eficiente que a LagClus.

1.4 Organização da Tese

Esta tese está organizada em sete capítulos, sendo este o primeiro. No Capítulo 2 são inicialmente apresentados os conceitos básicos sobre grafo de conflitos, particionamento de grafos, relaxação Lagrangeana e Geração de Colunas. Em seguida, no Capítulo 3 é apresentada a LagClus e o AGC aplicados a um Problema de Máximo Conjunto Independente de Vértices (PMCIIV).

No Capítulo 4 são apresentadas contribuições para o PRCP: dois novos modelos matemáticos e os resultados da LagClus e do AGC sobre instâncias do PRCP. Tanto a LagClus quanto o AGC encontraram as soluções ótimas de várias instâncias, superando os resultados até então obtidos na literatura.

O Capítulo 5 apresenta os resultados da LagClus e do AGC sobre instâncias do PCPP. Os resultados computacionais mostram que a LagClus e o AGC são métodos fortes de solução, fornecendo as soluções ótimas de várias instâncias consideradas difíceis na literatura.

Ainda no Capítulo 5 é apresentado o PEUC que aparece frequentemente nos portos brasileiros. Como será mostrado, devido às restrições operacionais, o PEUC pode ser considerado como sendo um PCPP, porém em escala maior. Neste caso, a LagClus foi utilizada em instâncias reais obtidas nos portos brasileiros e os seus resultados foram melhores que os usados na prática, proporcionando um aumento no número de unidades de celulose estivadas e, conseqüentemente, uma redução nos custos logísticos.

No Capítulo 6 são apresentados os experimentos computacionais com a LagClus para o PPDISO considerando instâncias do satélite SPOT 5 (<http://spot5.cnes.fr>). Esse problema não apresenta um grafo de conflitos bem definido e mesmo assim a LagClus forneceu bons resultados. Ainda nesse capítulo, são propostas várias formulações matemáticas para o PPDISO que permitem encontrar as soluções ótimas de instâncias propostas na literatura que até então eram desconhecidas. Essas formulações são baseadas na teoria de cortes.

Por último, no Capítulo 7 encontra-se um sumário das principais contribuições e conclusões, seguidas de sugestões para futuras pesquisas.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

O presente capítulo tem como objetivo apresentar os principais conceitos utilizados neste trabalho relacionados à relaxação lagrangeana, método de geração de colunas, teoria de planos de corte, grafos e particionamento de grafos.

Inicialmente, são apresentados conceitos gerais sobre relaxação que dão suporte à relaxação lagrangeana. Em seguida, as teorias do método de geração de colunas e de planos de corte são mostradas. Por último, são apresentados os conceitos de grafos e as técnicas de particionamento de grafos.

Todos esses conceitos formam a base da relaxação lagrangeana com *clusters* (LagClus) e do Algoritmo de Geração de Colunas (AGC), que serão apresentados no próximo capítulo.

2.1 Conceitos Sobre Relaxação

Toda a teoria aqui apresentada foi adaptada dos trabalhos de Parker e Rardin (1988), Wolsey (1998) e Guignard (2003).

Considere o seguinte problema:

$$v(P) = \text{Max } cx \quad (2.1)$$

(P) Sujeito a:

$$Ax \leq b \quad (2.2)$$

$$Dx \leq d \quad (2.3)$$

$$x \in Z_+^n \quad (2.4)$$

Nesse problema a função objetivo 2.1 deve ser maximizada, satisfazendo as restrições 2.2-2.4. Suponha agora que P é um problema “difícil” de ser resolvido devido ao conjunto de restrições 2.3, de tal maneira que, se esse conjunto for removido, o problema resultante (2.1-2.2, 2.4) é de “fácil” resolução.

De maneira geral, a relaxação de um problema tem como finalidade torná-lo mais “fácil”

de ser resolvido. Porém, ao se resolver um problema relaxado, não existem garantias de que a solução relaxada seja uma solução viável para o problema original. Uma solução é viável para um problema se satisfaz todas as suas restrições.

A solução de um problema relaxado gera um limitante para a solução ótima x^* do problema original que, se for de minimização (maximização), o valor da função objetivo do problema relaxado fornece um limitante inferior (superior) para a solução ótima do problema em questão.

De acordo com Parker e Rardin (1988) e Wolsey (1998), define-se como relaxação de um problema de maximização P , um problema RP tal que:

- a. Cada solução viável de P também é uma solução viável de RP ;
- b. O valor da função objetivo do problema P , avaliada para qualquer ponto x viável, é menor ou igual à função objetivo de RP para o mesmo ponto x .

A afirmação b) pode ser alterada para contemplar um problema de minimização. Neste caso, se P for um problema de minimização:

- c. O valor da função objetivo do problema P , avaliada para qualquer ponto x viável, é maior ou igual à função objetivo de RP para o mesmo ponto x .

Lema 2.1.1 *Seja P como definido em 2.1-2.4 e RP uma relaxação de P . Então $v(RP) \geq v(P)$.*

Prova: Qualquer ponto x viável em P é viável também em RP . Assim, se P não é viável ($v(P) = -\infty$) a desigualdade acima ($v(RP) \geq v(P)$) se mantém. Se $v(P)$ é ilimitado ($v(P) = +\infty$), então $v(RP)$ também é ilimitado. Se $v(P)$ tem um ótimo e este ótimo é finito, o valor da função objetivo de P , o qual é menor ou igual à função objetivo de RP , fornece um limitante inferior a $v(RP)$ pois a solução ótima de P é uma solução viável em RP .

Lema 2.1.2 *Seja P como definido em 2.1-2.4 e RP uma relaxação de P tal que rx seja a função objetivo de RP . Se x^* é a solução ótima de RP , é viável em P e $cx^* = rx^*$, então x^* é a solução ótima de P .*

Prova: Uma solução x^* que satisfaça as hipóteses do Lema 2.1.2 é, por hipótese, uma

solução viável em P . Isto implica dizer que $cx^* \leq v(P)$. Então, considerando o Lema 2.1.1, $rx^* = v(RP) \geq v(P) \geq cx^* = rx^*$. Segue então que x^* é uma solução ótima de P .

Considere novamente o problema P (2.1-2.4). Se o domínio x for alterado de tal maneira que $x_i \in \Re$ para todo i , então, tem-se uma relaxação de programação linear (PL) do problema P .

Lema 2.1.3 *Seja PL a relaxação de programação linear de P . As seguintes relações entre P e PL são válidas:*

- a. $v(P) \leq v(PL)$;
- b. *Se a solução x_{PL} do problema PL for tal que cada componente é não-negativa e inteira, então x_{PL} é também solução de P .*

Este Lema pode ser provado utilizando os Lemas 2.1.1 e 2.1.2.

2.2 Relaxação Lagrangeana

Considerando novamente o problema P como definido em 2.1-2.4, ao se remover as restrições 2.3, o problema relaxado resultante fornece um limitante superior para P , porém este limitante pode ser muito superior à solução ótima $v(P)$. Por outro lado, pode-se tentar reduzir esse limitante superior de tal maneira que este caminhe em direção a $v(P)$. Essa é a idéia da relaxação Lagrangeana.

A utilização da relaxação lagrangeana cresceu a partir dos trabalhos de Held e Karp (1970) e Held e Karp (1971) que a usaram para obter um limitante inferior para o problema do caixeiro viajante. Desde então, esse método vem sendo aplicado em diversos problemas.

2.2.1 Forma Geral e Dual Lagrangeano

Considere agora o problema P de forma mais geral:

$$v(P) = \text{Max } cx \tag{2.5}$$

(P) Sujeito a:

$$Dx \leq d \tag{2.6}$$

$$x \in X = \{x \in Z_+^n, Ax \leq b\} \quad (2.7)$$

O problema P acima também é conhecido como problema primal.

Para qualquer valor de $\lambda = (\lambda_1, \dots, \lambda_m) \geq 0$, o problema abaixo é uma relaxação de P.

$$v(L_\lambda P) = \text{Max } cx + \lambda(d - Dx) \quad (2.8)$$

($L_\lambda P$) Sujeito a 2.7.

Note que no problema relaxado $L_\lambda P$ as restrições complicadoras aparecem na função objetivo como um termo de penalidade $\lambda(d - Dx)$. Neste caso, λ são os multiplicadores lagrangeanos associados às restrições $Dx \leq d$. Assim, o problema $L_\lambda P$ é uma relaxação lagrangeana de P com parâmetro λ .

Como mostrado acima, se $L_\lambda P$ é uma relaxação de P, então $v(L_\lambda P)$ é um limitante superior de $v(P)$. Para encontrar o melhor limitante superior, ou seja, o menor possível, sobre uma infinidade de valores possíveis para λ , deve-se resolver o problema Dual Lagrangeano (DL):

$$v(DL) = \text{Min } v(L_\lambda P) \quad (2.9)$$

(DL) Sujeito a:

$$\lambda \geq 0 \quad (2.10)$$

Por outro lado, quando as m restrições a serem dualizadas são restrições de igualdade, da seguinte maneira $Dx = d$, os correspondentes multiplicadores lagrangeanos devem ser irrestritos ($\lambda \in \Re^m$), e o dual lagrangeano fica assim definido: $v(DL) = \text{Min}_\lambda v(L_\lambda P)$.

A resolução da relaxação lagrangeana $L_\lambda P$ pode levar à solução ótima do problema original P como mostra a Proposição 2.2.1.

Proposição 2.2.1 Se $\lambda \geq 0$,

- a) $x(\lambda)$ é uma solução ótima de $L_\lambda P$,
- b) $Dx(\lambda) \leq d$, e ainda

$$c) \quad (Dx(\lambda))_i = d_i \text{ sempre que } \lambda_i > 0,$$

então $x(\lambda)$ é a solução ótima de P .

Prova: De (a) $v(DL) \leq v(L_\lambda P) = cx(\lambda) + \lambda(d - Dx(\lambda))$. De (c) $cx(\lambda) + \lambda(d - Dx(\lambda)) = cx(\lambda)$. De (b) $x(\lambda)$ é viável em P , então $cx(\lambda) \leq v(P)$. Assim $v(DL) \leq cx(\lambda) + \lambda(d - Dx(\lambda)) = cx(\lambda) \leq v(P)$. Mas como $v(DL) \geq v(P)$, a igualdade entre $v(DL)$ e $v(P)$ se faz necessária e assim, $x(\lambda)$ é a solução ótima de P .

Note também que se as restrições dualizadas forem restrições de igualdade, a condição (c) é automaticamente satisfeita, e uma solução ótima de $L_\lambda P$ é também ótima em P se ela for viável em P .

2.2.2 Algoritmo de Subgradientes

Pela Proposição 2.2.1, pode-se tentar encontrar a solução ótima através da relaxação lagrangeana para um problema P . Porém, existe a necessidade de se escolher os multiplicadores lagrangeanos $\lambda \geq 0$ e resolver o dual lagrangeano não é uma tarefa simples, pois para cada valor de λ deve-se resolver uma relaxação lagrangeana $L_\lambda P$. Entretanto, $L_\lambda P$ em função do multiplicador lagrangeano λ , apresenta a propriedade de ser convexa e linear por partes.

Sendo assim, utiliza-se normalmente o conceito de subgradientes para aproximar, no sentido lagrangeano, as soluções de um problema relaxado à solução ótima (Parker e Rardin, 1988). A Figura 2.1 descreve os passos necessários para o algoritmo de subgradientes proposto por Parker e Rardin (1988).

Dois detalhes são importantes sobre o algoritmo de subgradientes. Primeiro, pode-se observar que o valor da melhor solução v^k é mantido no passo 2. Isso é necessário porque os passos do algoritmo de subgradientes não garantem melhorias em $L_\lambda P$. O outro detalhe diz respeito ao tamanho do passo π_k , que deve respeitar as condições descritas no passo 3 para que haja convergência. Parker e Rardin (1988) apresentam um exemplo e provam que essas condições são suficientes para garantir a convergência do método.

Na literatura existem vários trabalhos sobre relaxação lagrangeana. Uma boa revisão pode ser obtida em Shapiro (1974), Wolsey (1998), Parker e Rardin (1988) e Guignard (2003). O trabalho de Guignard (2003) mostra vários resultados intrigantes e questões a serem estudadas.

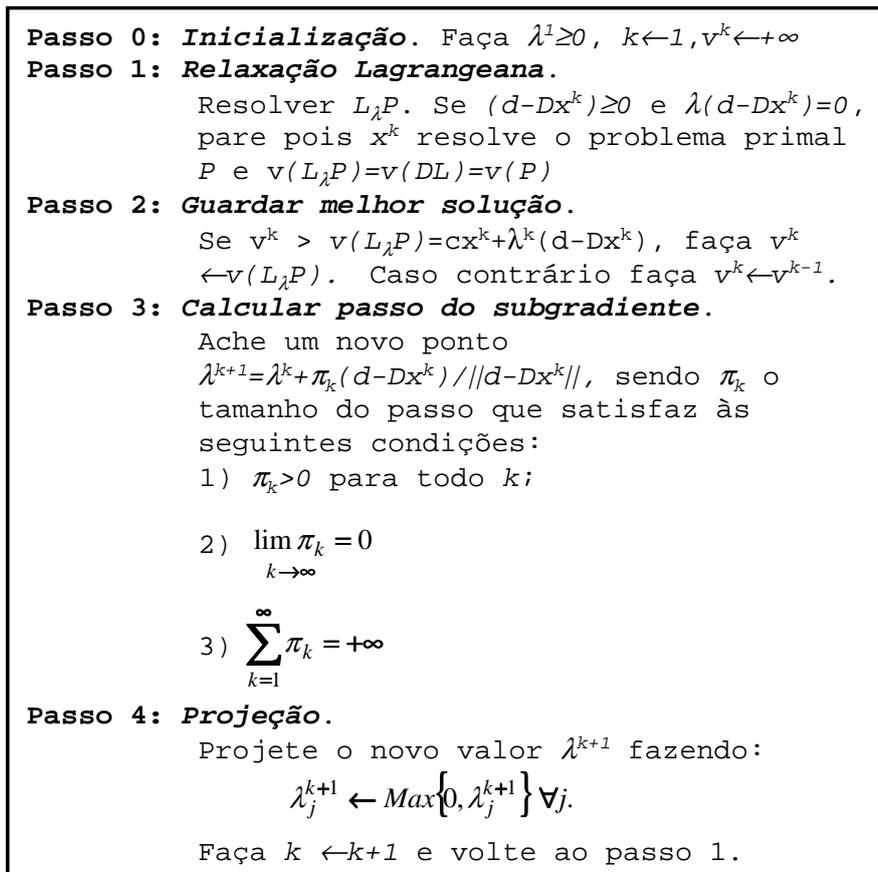


FIGURA 2.1 – Algoritmo de Subgradientes segundo Parker e Rardin (1988).

2.3 Decomposição Dantzig-Wolfe

Considere um problema $(P) \{v(P) = \text{Max } cx : x \in X\}$ tal que a região factível X pode ser escrita como uma interseção de dois ou mais conjuntos com a seguinte estrutura $X = \bigcap_{k=1}^K X^k$ para algum $K \geq 1$. Desta maneira, suponha um problema P com a seguinte representação:

$$v(P) = \text{Max} \sum_{k=1}^K c^k x^k \tag{2.11}$$

(P) Sujeito a:

$$+A^1 x^1 + A^2 x^2 \dots + A^K x^K \leq b \tag{2.12}$$

$$\begin{aligned} +D^1 x^1 &\leq d_1 \\ +D^2 x^2 &\leq d_2 \\ &\dots \leq \vdots \\ +D^K x^K &\leq d_K \end{aligned} \tag{2.13}$$

$$x^1 \in Z_+^{n_1} \quad x^2 \in Z_+^{n_2} \quad \dots \quad x^K \in Z_+^{n_K} \quad (2.14)$$

tal que os conjuntos $X^k = \{x^k \in Z_+^{n_k} : D^k x^k \leq d_k\}$ são independentes para $k = 1, \dots, K$ e somente o conjunto de restrições 2.12 conecta os diferentes conjuntos de restrições.

Considerando P descrito anteriormente e a relaxação lagrangeana apresentada na Seção 2.2, note que ao relaxar 2.12, o problema relaxado $L_\lambda P$ pode ser decomposto em K subproblemas $(L_\lambda P) \left\{ v(L_\lambda P) = \sum_{k=1}^K \text{Max} ((c^k - \lambda A^k)x^k : x^k \in X^k) + \lambda b \right\}$, uma vez que as restrições restantes são independentes entre si.

Problemas que apresentam as características do problema P, ou seja, formados por blocos de restrições independentes (ex: restrições 2.13) e por um único bloco de acoplamento (ex: restrições 2.12), podem ser explorados de uma outra maneira, aplicando a decomposição Dantzig-Wolfe (Dantzig e Wolfe, 1960).

Considerando que cada conjunto $X^k = \{x^k \in Z_+^{n_k} : D^k x^k \leq d_k\}$ possui finito conjunto T_k de pontos extremos $\{x^{k,t}\}_{t=1}^{T_k}$, cada ponto do conjunto X^k pode ser escrito como uma combinação linear convexa de seus pontos extremos. Sendo assim, X^k pode ser reescrito como:

$$X^k = \left\{ x^k \in \mathfrak{R}^{n_k} : x^k = \sum_{t=1}^{T_k} \alpha_{k,t} x^{k,t}, \sum_{t=1}^{T_k} \alpha_{k,t} = 1, \alpha_{k,t} \in [0, 1] \forall t = 1, \dots, T_k \right\} \quad (2.15)$$

Agora usando a definição de X^k descrita em 2.15, o problema P pode ser assim definido:

$$v(P) = \sum_{k=1}^K \sum_{t=1}^{T_k} (c^k x^{k,t}) \alpha_{k,t} \quad (2.16)$$

(P) Sujeito a:

$$\sum_{k=1}^K \sum_{t=1}^{T_k} (A^k x^{k,t}) \alpha_{k,t} \leq b \quad (2.17)$$

$$\sum_{t=1}^{T_k} \alpha_{k,t} = 1 \quad \forall k = 1, \dots, K \quad (2.18)$$

$$\alpha_{k,t} \in [0, 1] \quad \forall t = 1, \dots, T_k, \quad k = 1, \dots, K \quad (2.19)$$

Assim, o problema de encontrar soluções viáveis (extremos) para X^k é chamado de subproblema e o problema 2.16-2.19 é denominado de Problema Mestre (PM). Esses dois problemas definem a Decomposição de Dantzig-Wolfe.

2.3.1 O Método de Geração de Colunas na Decomposição Dantzig-Wolfe

Considerando o problema mestre 2.16-2.19 (PM_{PL}), seja $\begin{pmatrix} A^k x \\ 1 \end{pmatrix}$ uma coluna para cada $x \in X^k$. Sejam ainda $\{\pi_i\}_{i=1}^m$ as variáveis duais associadas às m restrições de acoplamento 2.17 e $\{\mu_i\}_{i=1}^K$ as variáveis duais associadas às restrições de convexidade 2.18.

O problema mestre requer que todos os pontos extremos de X^k sejam definidos, porém em geral, o número de pontos extremos é muito grande e encontrá-los inviabiliza o uso direto do problema mestre. Sendo assim, muito frequentemente opta-se por trabalhar com um subconjunto de pontos extremos (Barnhart et al., 1998), e em seguida, geram-se novos pontos extremos de maneira sistemática como mostrado abaixo no Algoritmo de Geração de Colunas (AGC).

Seja $X^{k,l} \subseteq X^k$ um subconjunto que representa l pontos extremos de X^k . O problema mestre 2.16-2.19 quando considera apenas um subconjunto de pontos extremos $X^{k,l}$ para cada k , passa a ser denominado Problema Mestre Restrito (PMR_{PL}).

Com isso, o PMR pode ser assim generalizado:

$$\tilde{v}(PMR_{PL}) = Max \quad \tilde{c}\tilde{\alpha} \quad (2.20)$$

(PMR_{PL}) Sujeito a:

$$\tilde{A}\tilde{\alpha} \leq \tilde{b} \quad (2.21)$$

$$\tilde{\alpha} \geq 0 \quad (2.22)$$

sendo $\tilde{b} = \begin{pmatrix} b \\ 1 \end{pmatrix}$, \tilde{A} uma matriz gerada conforme o número de colunas disponíveis descrita da seguinte maneira

$$\begin{bmatrix} A^1 x^{1,1} & \dots & A^1 x^{1,T_1} & A^2 x^{2,1} & \dots & A^2 x^{2,T_2} & \dots & A^K x^{K,1} & \dots & A^K x^{K,T_K} \\ 1 & \dots & 1 & & & & & & & \\ & & & 1 & \dots & 1 & & & & \\ & & & & & & \dots & & & \\ & & & & & & & 1 & \dots & 1 \end{bmatrix}, \text{ e } \tilde{c} \text{ e}$$

\tilde{a} os correspondentes custos e variáveis.

Ao resolver o PMR_{PL} , obtém-se uma solução ótima \tilde{a}^* do problema e uma solução ótima do dual $(\pi, \mu) \in \Re^m \times \Re^K$. Qualquer solução viável do PMR_{PL} é viável para PM_{PL} . Em particular, se \tilde{a}^* é uma solução viável de PMR_{PL} , então $\tilde{v}(\text{PMR}_{PL}) = \tilde{c}\tilde{a}^* = \sum_{i=1}^m \pi_i b_i + \sum_{k=1}^K \mu_k \leq v(\text{PM}_{PL})$ (Wolsey, 1998).

Com o objetivo de inserir novas colunas que melhorem a função objetivo do PMR_{PL} , o subproblema passa a ser denominado de subproblema gerador de colunas, pois as variáveis, obtidas através da sua resolução, constituirão novas colunas para o PMR_{PL} . O subproblema pode ser assim definido:

$$v(\zeta)^k = \text{Max} \{ (c^k - \pi A^k)x - \mu_k : x \in X^k \} \forall k = 1, \dots, K \quad (2.23)$$

Desta forma, o subproblema torna-se um problema de custo reduzido ou problema *pricing*.

Considere $\zeta_k = v(\zeta)^k$. Se $\zeta_k > 0$ para algum k , a solução correspondente à solução ótima \tilde{x}^k do subproblema tem um custo reduzido positivo e a coluna $\begin{pmatrix} A^k \tilde{x}^k \\ 1 \end{pmatrix}$ deve ser inserida no PMR_{PL} e o novo PMR deve ser novamente resolvido.

2.3.2 Algoritmo Básico de Geração de Colunas

Considerando o subproblema, tem-se que $\zeta_k \geq (c^k - \pi A^k)x - \mu_k$ para todo $x \in X^k$. Isto implica em $(c^k - \pi A^k)x - \mu_k - \zeta_k \leq 0$ para todo $x \in X^k$. Assim, sendo $\zeta = (\zeta_1, \dots, \zeta_K)$, tem-se que $(\pi, \mu + \zeta)$ é um dual viável para o PM_{PL} . Portanto:

$$v(\text{PM}_{PL}) \leq \pi b + \sum_{k=1}^K (\mu_k + \zeta_k) \quad (2.24)$$

A relação acima, leva diretamente ao Algoritmo de Geração de Colunas (AGC) que termina quando $\zeta_k = 0$ para todo $k = 1, \dots, K$. A Figura 2.2 apresenta um esquema básico do AGC.

Admitindo-se que são evitadas a degeneração do subproblema e as soluções cíclicas no problema mestre, o algoritmo converge em um número finito de iterações, pois consiste no algoritmo Simplex Revisado com a inclusão dos custos reduzidos dados pela resolução do subproblema.

<p>Passo 0: Inicialização. Determine um conjunto viável inicial de colunas para o PMR. Faça $i \leftarrow 1$</p> <p>Passo 1: Resolva o PMR_{PL}.</p> <p>Passo 2: Resolva o subproblema gerador de colunas. Com os valores duais (π, μ) do PMR_{PL} obtidos no Passo 2, atualize e resolva o subproblema.</p> <p>Passo 3: Adicione colunas no PMR_{PL}. Adicione as novas colunas com custo reduzido positivo no PMR_{RL}, faça $i \leftarrow i+1$ e volte ao Passo 1.</p> <p>Se não existirem novas colunas pare pois a solução é ótima.</p>

FIGURA 2.2 – Algoritmo básico de Geração de Colunas.

Para uma boa revisão sobre Geração de Colunas bem como aplicações, veja Wolsey (1998), Barnhart et al. (1998), Desrosiers e Lübbecke (2005), Carvalho (2005), Pereira et al. (2007) e Senne et al. (2007).

2.4 Planos de Corte

Nesta seção são apresentados os conceitos básicos sobre os métodos de planos de corte. Esses métodos foram inicialmente introduzidos por Gomory (1958) para resolver problemas de programação linear inteira. Eles se baseiam na seguinte idéia: começa-se com uma solução relaxada de PL de um problema P, depois, de maneira sistemática, enquanto a solução ótima não for inteira, procura-se eliminá-la acrescentando-se novas desigualdades (restrições), conhecidas como planos de corte, ao problema linear corrente.

Os planos de corte podem ajudar a fortalecer uma formulação, de tal maneira que soluções inteiras sejam obtidas. Porém, definir os cortes mais importantes é uma tarefa complicada. Por outro lado, dadas duas formulações diferentes F^1 e F^2 de um problema P, é possível responder a seguinte pergunta: qual destas formulações é a mais eficiente? Isto é, qual delas encontra a solução ótima em um menor tempo? A resposta está na definição do *Poliedro* do problema que é definido pelas restrições de P.

2.4.1 Poliedros

Considere a seguinte formulação de um problema (P) dada por $\{v(P) = \text{Max}(cx : Dx \leq d, x \in Z_+^n)\}$. O poliedro \mathbf{P} do problema P é um subconjunto de \mathfrak{R}^n descrito por um conjunto finito de restrições lineares $\mathbf{P} = \{x \in \mathfrak{R}^n : Dx \leq d\}$. Um poliedro \mathbf{P} definido por uma matriz D e por um vetor d é denotado por $\mathbf{P}(D, d)$. Se \mathbf{P} é um poliedro limitado, ou seja, existe $\alpha \in \mathfrak{R}$ tal que todo ponto $x \in \mathbf{P}$ satisfaz $\|x\| \leq \alpha$, então \mathbf{P} é chamado *politopo*.

Se $a \in \mathfrak{R}^n \setminus 0$ e $\alpha \in \mathfrak{R}$, então o poliedro $\{x \in \mathfrak{R}^n : ax \leq \alpha\}$ é chamado *semi-espaço*. Um conjunto de pontos que divide o espaço em dois semi-espaços, ou seja, um conjunto da forma $\{x \in \mathfrak{R}^n : ax = \alpha\}$, em que a é não nulo, é chamado de *hiperplano*. Um conjunto vazio é um poliedro, assim como o espaço \mathfrak{R}^n .

Observando que cada uma das linhas da matriz D juntamente com a correspondente entrada do vetor d define um semi-espaço, segue imediatamente que $\mathbf{P}(D, d)$ é a intersecção de um número finito de semi-espaços.

Por exemplo, na Figura 2.3 a região mais escura corresponde ao poliedro $\mathbf{P}(D, d)$ onde

$$D = \begin{pmatrix} -1 & 1 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \text{ e } d = \begin{pmatrix} 2 \\ -2 \\ 4 \\ 4 \end{pmatrix}.$$

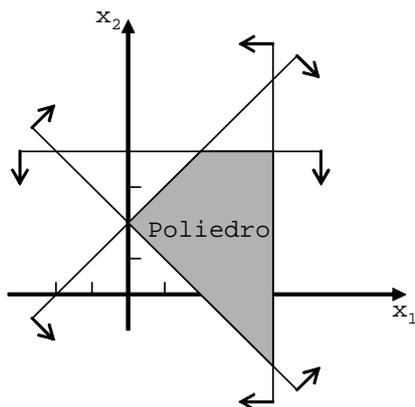


FIGURA 2.3 – Exemplo de um poliedro no \mathfrak{R}^n .

Nem sempre os poliedros surgem na forma de um sistema de inequações. Entretanto, é possível mostrar que sistemas constituídos por equações e inequações podem ser transformados em sistemas equivalentes somente constituídos de inequações.

2.4.2 Formulação “Ideal”

Como mostrado por Wolsey (1998), suponha que para um mesmo problema inteiro P existem três formulações diferentes F^1 , F^2 e F^3 . Qual delas é a melhor formulação para P ? A resposta pode ser obtida geometricamente.

Considere que os poliedros mostrados na Figura 2.4 são os das formulações F^1 , F^2 e F^3 . Nesse caso, a formulação F^3 é a ideal porque se a sua relaxação de PL for resolvida, a solução ótima será um dos extremos do poliedro P^3 e como todos são inteiros, o problema P está resolvido. Veja novamente Lemas 2.1.2 e 2.1.3.

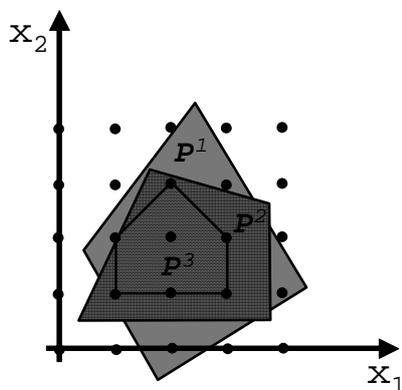


FIGURA 2.4 – Poliedros das formulações F^1 , F^2 e F^3 .

Agora a seguinte definição pode ser apresentada.

Definição 2.4.1 Dado um conjunto $X \subseteq \mathbb{R}^n$ e duas formulações F^1 e F^2 para um mesmo problema P , e sejam P^1 e P^2 os poliedros relativos às formulações F^1 e F^2 , respectivamente. F^1 é uma formulação melhor que F^2 se $P^1 \subset P^2$ e $P^1 \neq P^2$.

Considere novamente um problema $(P) \{v(P) = \text{Max}(cx : Dx \leq d, x \in Z_+^n)\}$.

Definição 2.4.2 Dado um conjunto $X \subseteq \mathbb{R}^n$, a envoltória convexa de X representada por $\text{Conv}(X)$, é definida como: $\text{conv}(X) = \left\{ \begin{array}{l} x : x = \sum_{i=1}^t \alpha_i x^i, \sum_{i=1}^t \alpha_i = 1, \alpha_i \geq 0 \text{ para } i = 1, \dots, t \\ \text{onde } x^i \in X \text{ para } i = 1, \dots, t \end{array} \right\}$ (Wolsey, 1998).

Proposição 2.4.3 $\text{Conv}(X)$ é um Poliedro.

Proposição 2.4.4 *Os pontos extremos da $Conv(X)$ estão em X .*

As provas das proposições 2.4.3 e 2.4.4 não são aqui apresentadas. Para demais detalhes, ver Parker e Rardin (1988) e Ferreira e Wakabayashi (1996).

As duas proposições acima, permitem substituir P pelo equivalente problema de programação linear: $(P_{PL}) \{v(P_{PL}) = \text{Max}(cx : x \in Conv(X))\}$. Isso também é válido para problemas mistos (com variáveis inteiras e reais).

O problema da equivalência mostrada acima é enumerar todas as inequações necessárias para descrever $Conv(X)$. Normalmente o número dessas inequações é enorme inviabilizando o processo (Wolsey, 1998), porém dado uma instância de um problema, é possível obter inequações válidas que aproximam a $Conv(X)$ para a dada instância.

Definição 2.4.5 *Uma desigualdade $\pi x \leq \pi_0$, $x \in \mathfrak{R}^n$, é uma inequação válida para $X \subseteq \mathfrak{R}^n$ se $\pi x \leq \pi_0$ para todo $x \in X$.*

Se $X = \{x \in Z^n : Dx \leq d\}$ e $Conv(X) = \{x \in \mathfrak{R}^n : \tilde{D}x \leq \tilde{d}\}$, as restrições $D^i x \leq d_i$ e $\tilde{D}^i x \leq \tilde{d}_i$ são inequações válidas para X .

2.4.3 Algumas Desigualdades Válidas

Para Problemas Lineares

Dado um poliedro $\mathbf{P} = \{x : Dx \leq d, x \geq 0\}$, uma desigualdade $\pi x \leq \pi_0$ é válida para \mathbf{P} se algumas condições forem satisfeitas.

Proposição 2.4.6 *$\pi x \leq \pi_0$ é válida para $\mathbf{P} \neq \emptyset$ se, e somente se*

- a) *existe $u \geq 0$ e $v \geq 0$ tal que $uD - v = \pi$ e $ud \leq \pi_0$, ou*
- b) *existe $u \geq 0$ tal que $uD \geq \pi$ e $ud \leq \pi_0$.*

Prova: De (b), se existe $u \geq 0$ tal que $uD \geq \pi$ e $ud \leq \pi_0$, então para qualquer $x \in \mathbf{P}$, $Dx \leq d \Leftrightarrow uDx \leq ud \Leftrightarrow \pi x \leq uDx \leq ud \leq \pi_0 \Leftrightarrow \pi x \leq \pi_0$ é uma desigualdade válida.

Para Problemas Inteiros

O procedimento de Chvátal-Gomory, para construir inequações válidas, vem sendo utilizado em aplicativos comerciais de otimização como no CPLEX (ILOG, 2001a) e

no XPRESS (XPRESS, 2006). Este procedimento permite gerar desigualdades válidas para um problema inteiro.

Seja $X = \mathbf{P} \cap Z^n$ onde $\mathbf{P} = \{x \in \mathfrak{R}_+^n : Dx \leq d\}$, D uma matriz $m \times n$ com colunas $\{D^1, D^2, \dots, D^n\}$ e $u \in \mathfrak{R}_+^m$:

- a) a desigualdade $\sum_{j=1}^n uD^j x_j \leq ud$ é válida para \mathbf{P} pois $u \geq 0$ e $\sum_{j=1}^n D^j x_j \leq d$,
- b) a desigualdade $\sum_{j=1}^n \lfloor uD^j \rfloor x_j \leq ud$ é válida para \mathbf{P} pois $u \geq 0$,
- c) a desigualdade $\sum_{j=1}^n \lfloor uD^j \rfloor x_j \leq \lfloor ud \rfloor$ é válida para X pois x é inteiro e assim $\sum_{j=1}^n \lfloor uD^j \rfloor x_j$ é inteiro.

Seja $X = X^1 \cup X^2$ sendo $X^i \subseteq \mathfrak{R}_+^n$, uma disjunção (união) de dois conjuntos X^1 e X^2 . Alguns resultados importantes podem ser obtidos.

Proposição 2.4.7 *Se $\sum_{j=1}^n \pi_j^i x_j \leq \pi_0^i$ é uma desigualdade válida para X^i , $i = 1, 2$, então a desigualdade $\sum_{j=1}^n \pi_j x_j \leq \pi_0$ é válida para X se $\pi_j \leq \text{Min}\{\pi_j^1, \pi_j^2\}$ para $j = 1, \dots, n$ e $\pi_0 \geq \text{Max}\{\pi_0^1, \pi_0^2\}$.*

Prova: Se $x \in X$, então $x \in X^1$ ou $x \in X^2$. Se $x \in X^i$, então como $x \geq 0$, $\sum_{j=1}^n \pi_j x_j \leq \sum_{j=1}^n \pi_j^i x_j \leq \pi_0^i \leq \pi_0$ para $i = 1, 2$. Assim, a desigualdade é válida para todo $x \in X$.

Proposição 2.4.8 *Se $\mathbf{P}^i = \{x \in \mathfrak{R}_+^n : D^i x \leq d^i\}$, para $i = 1, 2$, são poliedros não vazios, então $\sum_{j=1}^n \pi_j x_j \leq \pi_0$ é uma desigualdade válida para $\text{Conv}(\mathbf{P}^1 \cup \mathbf{P}^2)$ se, e somente se existem $u_1, u_2 \geq 0$ tal que: $\pi \leq u_1 D^1$, $\pi \leq u_2 D^2$, $\pi_0 \geq u_1 d^1$ e $\pi_0 \geq u_2 d^2$.*

Para prova, veja Wolsey (1998).

Muitos outros cortes dependem das características do problema em questão. Neste caso, têm-se cortes específicos que não serão aqui apresentados, como os cortes para o problema da Mochila ou de Localização de Facilidades. Cortes específicos para problemas binários modelados em grafos, são apresentados na próxima seção.

2.5 Representação em Grafos

Um grafo $G = (V, A)$ consiste de um conjunto finito não vazio V de elementos denominados vértices e um conjunto de pares não ordenados de elementos distintos de V , denominado arestas, isto é, $A \subseteq \{(u, v) : u, v \in V, u \neq v\}$. Note que aqui não são permitidos “laços” e nem arestas “múltiplas”. A Figura 2.5 apresenta um grafo simples.

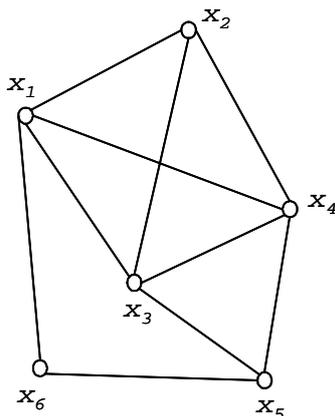


FIGURA 2.5 – Exemplo de um grafo $G = (V, A)$.

Existem vários problemas de OC modelados sobre grafos. O Problema do Caixeiro Viajante (PCV), por exemplo, bem difundido na literatura, pode ser representado por meio de um grafo em que o conjunto de vértices V representa as cidades que o caixeiro deve visitar e o conjunto de arestas A , os caminhos que ligam cada par de cidades. Note que neste caso, cada aresta tem uma distância associada (genericamente denominada peso) que é utilizada para o cálculo da distância total percorrida. Grafos que possuem pesos nas arestas são conhecidos como grafos ponderados.

2.5.1 Grafo de Conflitos

Conforme Atamtürk et al. (2000), um grafo de conflitos $G = (V, A)$ é um tipo especial de grafo em que os vértices representam variáveis lógicas binárias (por exemplo, de um problema de OC), ou seja, $V = \{x \in \{0, 1\}^n\}$, e as arestas representam suas relações lógicas tal que $x_i + x_j \leq 1 \forall (i, j) \in A$.

Uma aresta no grafo de conflitos indica que suas variáveis binárias (os vértices da aresta) não podem assumir o valor 1 simultaneamente. Essas arestas são conhecidas como conflitos ou arestas de adjacência. O exemplo mostrado na Figura 2.5 pode ser considerado um grafo de conflitos se os vértices representarem variáveis binárias e as arestas, incompatibilidade entre vértices.

O Problema do Máximo Conjunto Independente de Vértices (PMCIIV) está diretamente ligado aos grafos de conflitos. Esse problema tem como objetivo obter o maior subconjunto de vértices tal que todos eles sejam independentes entre si, ou seja, não apresente nenhuma aresta de conflito. Utilizando a teoria dos grafos, esse problema pode ser formalmente definido como: obter a cardinalidade do maior subgrafo de G que seja totalmente desconexo (Harary, 1972).

O PMCIIV é um problema clássico, bem difundido na literatura. Ele aparece freqüentemente embutido em várias aplicações como em modelos de alocação (Gerrard e Church, 1996), *Anti-Covering* (Murray e Church, 1997a), carregamento de paletes (Dowland, 1987b; Alvarez-Valdes et al., 2005a), planejamento de cortes florestais e de colheitas (Church et al., 1998; Goycoolea et al., 2005; Constantino et al., 2007), redes de telefonia celular (Björkund et al., 2005) e rotulação cartográfica (Zoraster, 1990; Yamamoto e Lorena, 2005).

O modelo matemático do PMCIIV pode ser assim definido (considere $G = (V, A)$ o seu grafo de conflitos):

$$v(\text{PMCIIV}) = \text{Max} \left\{ \sum_{v \in V} x_v \right\} \quad (2.25)$$

(PMCIIV) Sujeito a:

$$x_i + x_j \leq 1 \forall (i, j) \in A \quad (2.26)$$

$$x_i \in \{0, 1\} \forall i \in V \quad (2.27)$$

Se $x_i = 1$, o vértice i está incluído no conjunto independente, caso contrário, $x_i = 0$. O conjunto de restrições definido em 2.26 garante que dois vértices adjacentes não podem estar presentes simultaneamente na solução ótima do PMCIIV. Já o conjunto de restrições definido em 2.27 indica que todas as variáveis são binárias.

A Figura 2.6 mostra o grafo de conflitos de um PMCIIV, sua formulação matemática e sua solução ótima representada pelos vértices negros. Note que não é possível selecionar mais algum vértice sem violar uma restrição de adjacência.

Entretanto, em muitos problemas como os citados mais acima, o número de arestas presentes no grafo G é muito grande, o que pode inviabilizar o processo de obtenção da

solução ótima. O PMCIIV é classificado na literatura como *NP-Hard* (Harary, 1972).

Por outro lado, muitas técnicas estão presentes na literatura para fortalecer a formulação do PMCIIV através de planos de corte. Nemhauser e Sigismondi (1992) apresentam métodos que permitem aproximar o poliedro de uma instância do PMCIIV.

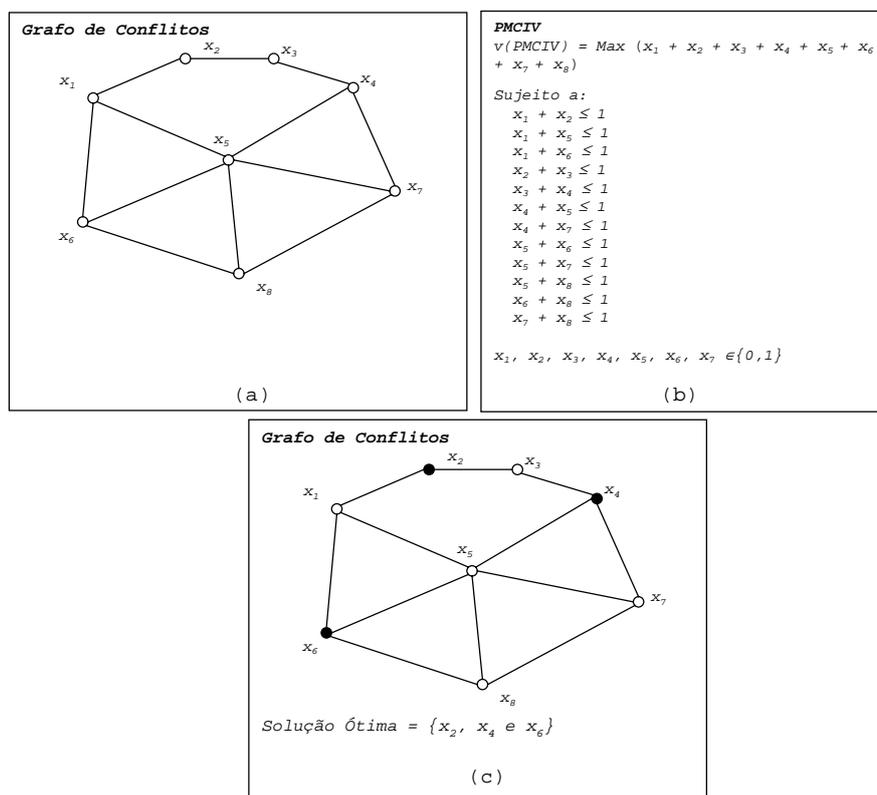


FIGURA 2.6 – PMCIIV. (a) Grafo de conflitos, (b) formulação matemática e (c) solução ótima do problema.

A seguir são demonstradas algumas desigualdades válidas para o PMCIIV. Considere $G = (V, A)$ o grafo de conflitos de uma instância do PMCIIV e \mathbf{P} o poliedro de uma instância associada ao máximo conjunto independente.

2.5.2 Desigualdades Válidas para o PMCIIV

Cliques

Dado $C \subseteq V$ uma clique em G , qualquer conjunto independente pode ter no máximo um vértice de C . Portanto, a desigualdade de clique:

$$x(C) \leq 1 \tag{2.28}$$

é uma desigualdade válida para \mathbf{P} . Padberg (1973) demonstrou que cada desigualdade do tipo 2.28 gera um corte em \mathbf{P} se, e somente se C é uma clique.

No exemplo da Figura 2.6, note que x_1, x_5 e x_6 formam uma clique. Neste caso, a restrição $x_1 + x_5 + x_6 \leq 1$ é uma restrição válida e as restrições $x_1 + x_5 \leq 1$, $x_1 + x_6 \leq 1$ e $x_5 + x_6 \leq 1$ podem ser removidas do modelo porque a restrição de clique as domina.

Odd Roles (Ciclos Ímpares) de Padberg (1973)

Um ciclo ímpar em G é um ciclo que contém um número ímpar de vértices. Se H é um ciclo ímpar em G , então a desigualdade associada ao ciclo ímpar H é:

$$x(V(H)) \leq \lfloor |V(H)| / 2 \rfloor. \quad (2.29)$$

Veja novamente o exemplo mostrado na Figura 2.6. Note que o conjunto $H = \{x_1, x_2, x_3, x_4, x_6, x_7, x_8\}$ é um ciclo ímpar em G . Neste caso, a restrição $x_1 + x_2 + x_3 + x_4 + x_6 + x_7 + x_8 \leq \lfloor 7/2 \rfloor$ é válida para o problema, mas não substitui nenhuma das restrições mostradas na Figura 2.6, diferentemente das restrições baseadas em cliques. Uma boa revisão sobre este assunto pode ser obtida em Cornuejols e Thizy (1982).

Rodas de Cheng e Cunningham (1997)

Outra classe de desigualdades mais recentes são as chamadas Rodas de Cheng e Cunningham (1997). Seja k um inteiro positivo e seja $G_1 = (V_1, A_1)$ um grafo com $V_1 = \{v_0, v_1, v_2, \dots, v_{2k+1}\}$ e $A_1 = \{(v_0, v_i), (v_i, v_{i+1}) : 1 \leq i \leq 2k+1\}$ tal que $v_{2k+2} = v_1$. Considere um subgrafo de G_1 . Sejam $P_{0,i}$ e $P_{i,i+1}$ os caminhos desde (v_0, v_i) e (v_i, v_{i+1}) respectivamente através da subdivisão. Este grafo é um *1-roda simples* (*simple 1-wheel*) de tamanho $2k+1$ se o ciclo C_i formado por $P_{0,i}$, $P_{i,i+1}$ e $P_{0,i+1}$ é ímpar para cada i . Esta *1-roda simples* pode ser denotada por $W = W(v_0; v_1, v_2, \dots, v_{2k+1})$ sendo v_0 denominado de centro (*hub*), $P_{0,1}, P_{0,2}, \dots, P_{0,2k+1}$ os raios (*spokes*), $P_{1,2}, P_{2,3}, \dots, P_{2k,2k+1}, P_{2k+1,1}$ os segmentos de borda (*rim-paths*), $v_1, v_2, \dots, v_{2k+1}$ as terminações dos raios (*spoke-ends*) e o ciclo formado por $P_{1,2}, P_{2,3}, \dots, P_{2k+1,1}$ a borda (*rim*). A Figura 2.7 apresenta um exemplo de uma *1-roda simples* sendo a o centro e b, c, d, g, i o conjunto de terminações dos raios.

Se os vértices forem divididos em dois conjuntos $\xi = \xi(W)$ e $O = O(W)$ onde $v_i \in \xi(O)$ se $P_{0,i}$ é um caminho par (ímpar). Na Figura 2.7, $O = d, g$ e $\xi = b, c, i$. Seja $S = S(W)$ o conjunto de vértices internos dos raios e $R = E(W)$ o conjunto de vértices dos segmentos de borda. Na Figura 2.7 $S = \{i, k, l\}$ e $R = \{m, e, f, h\}$.

Com este tipo de grafo, as seguintes desigualdades são válidas:

$$kx_0 + \sum_{i=1}^{2k+1} x_i + \sum_{v \in \xi} x_v + \sum_{v \in SUR} x_v \leq k + \frac{1}{2} (|S| + |R| + |\xi|) \quad (2.30)$$

$$(k+1)x_0 + \sum_{i=1}^{2k+1} x_i + \sum_{v \in O} x_v + \sum_{v \in SUR} x_v \leq k + \frac{1}{2} (|S| + |R| + |\xi| + 1) \quad (2.31)$$

Considerando o subgrafo da Figura 2.7, as seguintes restrições válidas podem ser geradas:

- $2x_a + 2x_b + 2x_c + 2x_i + \sum_{v \notin \{a,b,c,i\}} x_v \leq 7$; e
- $3x_a + 2x_d + 2x_g + \sum_{v \notin \{a,d,g\}} x_v \leq 7$.

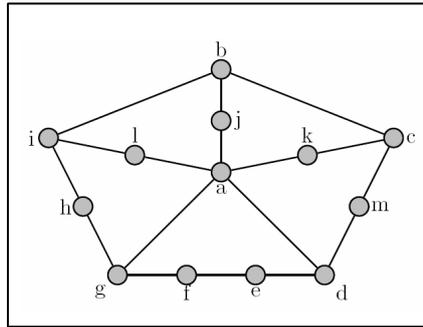


FIGURA 2.7 – Exemplo de 1-roda simples (Cheng e Cunningham, 1997).

Existem procedimentos baseados em caminhos mínimos entre pares de vértices que encontram as desigualdades de roda, mas, de acordo com Torres (2004), necessitam de um alto tempo computacional para serem executados.

Outras desigualdades válidas

Outras classes de desigualdades são os *antiholes* e *webs* de Balas e Padberg (1976) e *grilles* de Cánovas et al. (2000). Entretanto, não se conhecem procedimentos eficientes para obter esses cortes (Torres, 2004).

2.6 Particionamento de Grafos

Muitos problemas combinatoriais podem ser descritos como o problema de encontrar um ou mais conjuntos de vértices de um dado grafo, minimizando uma função objetivo que está relacionada às arestas que conectam os subconjuntos. Tais problemas incluem, por

exemplo, armazenamento eficiente em grandes bancos de dados (Shekhar e Liu, 1996) e mineração de dados (Cooley et al., 1997; Karypis e Kumar, 1999).

Genericamente, esse problema pode ser expresso como: dado um grafo ponderado $G = (V, A)$ no qual vértices e arestas possuem pesos associados, e um número \bar{P} , encontre \bar{P} subconjuntos $V_1, V_2, \dots, V_{\bar{P}}$ de V tal que:

- a) $\bigcup_{i=1}^{\bar{P}} V_i = V$, $V_i \cap V_j = \emptyset$ para todo $i \neq j$, e $V_i \neq \emptyset$ para todo $i = 1, \dots, \bar{P}$;
- b) $W(i) \approx W/\bar{P}$, $\forall i = 1, 2, \dots, \bar{P}$ onde $W(i)$ e W representam, respectivamente, as somas dos pesos dos vértices pertencentes a V_i e V ;
- c) O “tamanho do corte”, isto é, a soma dos pesos das arestas que conectam os subconjuntos V_i ($i = 1, 2, \dots, \bar{P}$) deve ser mínimo.

Qualquer conjunto $\{V_i \subseteq V : 1 \leq i \leq \bar{P}\}$ é chamado de uma partição se satisfaz a condição (a), ou seja, cada V_i é então uma parte de um \bar{P} -particionamento. Uma bissecção é um 2-particionamento e um particionamento que satisfaz a condição (b) é um particionamento balanceado.

O Problema de Particionamento de Grafos (PPG) é classificado como *NP-Hard* (Garey et al., 1976). Algoritmos capazes de encontrar a solução ótima de um PPG (ver, por exemplo, o algoritmo de Karisch et al. (1997)), são indicados para grafos pequenos, não sendo aplicáveis em problemas de grande porte (Fjällström, 1998).

Existem grafos que são particionados mais de uma vez devido às características do problema. Quando em um particionamento, tanto V quanto A não se alteram, diz-se que se trata de um caso estático e algoritmos conhecidos como seqüenciais são normalmente utilizados. No entanto, em alguns casos o grafo se altera de forma incremental, por exemplo, em métodos de elementos finitos adaptativos os vértices e arestas podem ser removidos ou adicionados durante os cálculos (Fjällström, 1998). Nesses casos, tem-se um caso dinâmico, e o particionamento de grafos é feito repetidamente. Com isso, segundo Fjällström (1998) e Karypis e Kumar (1998c), os algoritmos mais indicados para este problema são aqueles que fazem o particionamento paralelo, aproveitando informações dos particionamentos anteriores.

2.6.1 Algoritmos Seqüenciais

Pesquisadores têm desenvolvido muitos algoritmos seqüenciais para o PPG que são baseados em métodos de melhoramentos locais que, primeiramente, bisseccionam o grafo, e tentam reduzir o tamanho do corte, usando algum método de busca local.

Esses algoritmos são aplicados de forma recursiva para obter um \bar{P} -particionamento. Os algoritmos mais citados na literatura são o de Kernighan e Lin (1970), nomeado de algoritmo KL, o *Simulated Annealing* de Johnson et al. (1989), a Busca Tabu de Rolland et al. (1996), e o Algoritmo Genético de Bui e Moon (1996).

2.6.2 Algoritmos Globais

Um método global toma como entrada um grafo G e um inteiro \bar{P} , e gera uma \bar{P} -partição. Muitos desses métodos são recursivos, ou seja, eles bisseccionam o grafo e em cada subgrafo gerado, é repetido este processo até encontrar o número \bar{P} de partições desejadas. No entanto, esses métodos são aperfeiçoados com o uso de métodos de melhoramentos locais que procuram obter conjuntos independentes de vértices.

Os métodos classificados como globais ainda apresentam variações que dependem do modo como atuam no particionamento. Muitos grafos apresentam uma localização geométrica (coordenadas), assim, muitos algoritmos utilizam essa informação para realizar o particionamento, sendo estes classificados como algoritmos de particionamento geométricos (Miller et al., 1993). Outros grafos não apresentam essa propriedade e os métodos de particionamentos são classificados como métodos sem coordenação, levando então em consideração, a estrutura combinatorial do grafo. Dentre estes, os mais importantes são os métodos que utilizam em seu particionamento procedimentos para obtenção do maior subconjunto independente de vértices em cada subgrafo gerado.

Um algoritmo interessante apresentado na literatura é a bissecção recursiva de Simon (1991). Esse método inicia localizando dois vértices que possuem o maior caminho mínimo entre si, e em seguida um deles é selecionado. Usando busca em largura, a distância desse vértice em relação aos demais é determinada. Finalmente, os vértices são ordenados em relação a essa distância, e o conjunto ordenado é dividido em dois subconjuntos de mesmo tamanho, e assim sucessivamente.

Outra técnica que vem sendo empregada frequentemente é a bissecção recursiva em multi-níveis (Karypis e Kumar, 1998a,b). Em uma seção mais a frente, esse método será apresentado com detalhes.

2.6.3 Algoritmos Paralelos

Os algoritmos paralelos para o PPG, são implementações paralelas dos algoritmos apresentados nas seções anteriores. Alguns desses métodos, em particular os métodos geométricos, segundo (Fjällström, 1998), são relativamente simples de serem paralelizados.

A idéia por trás desses algoritmos é dividir a tarefa de particionamento entre diversos processadores, e com isso, ganhar em tempo computacional, permitindo trabalhar com grafos maiores.

Dentre os trabalhos mais citados estão o de Gilbert e Zmijewski (1987) que é baseado no algoritmo KL, o JOSTLE-D (Walshaw et al., 1997; Walshaw, 2004) e o algoritmo de reparticionamento paralelo e seqüencial de Schloegel et al. (1997).

2.6.4 Algoritmos de Multi-Níveis

Os algoritmos de particionamento de grafos em multi-níveis têm se tornado o método padrão para particionar grafos grandes e irregulares. Esses métodos vêm apresentando soluções de boa qualidade em um baixo tempo computacional (Karypis, 2002; Abou-Rjeili e Karypis, 2006).

Os algoritmos de multi-níveis, ao invés de tentarem realizar o particionamento diretamente no grafo, fazem primeiro, sucessivas aproximações do grafo original. Em cada uma dessas aproximações, um novo problema (um novo grafo) é obtido, menor que o anterior.

Esse processo continua até que uma aproximação gere um grafo pequeno com somente algumas dezenas de vértices. Nesse momento, o algoritmo realiza o particionamento do grafo final. Dado que o último grafo obtido é pequeno, heurísticas eficientes, como a KL, ou até mesmo métodos exatos, podem ser aplicados.

O passo final desses algoritmos é tomar o particionamento obtido com o menor grafo gerado, e propagá-lo em direção ao problema original. Essa propagação é realizada através de sucessivas aproximações do particionamento, seguidas de métodos de refinamento (Karypis e Kumar, 1998a,b; Karypis, 2002).

Na terminologia dos algoritmos de multi-níveis, o processo acima pode ser dividido em três fases:

- a) *Coarsening*;
- b) Particionamento; e
- c) *Uncoarsening* com refinamento.

Considere a Figura 2.8, note que as três fases estão mostradas. Na fase de *coarsing*, o grafo é reduzido sistematicamente, em seguida, na fase de particionamento, o grafo é

particionado em \bar{P} partições (na figura tem-se um 6-particionamento), e por último, na fase de *uncoarsening*, o particionamento obtido é refinado sucessivamente e projetado no particionamento do grafo original.

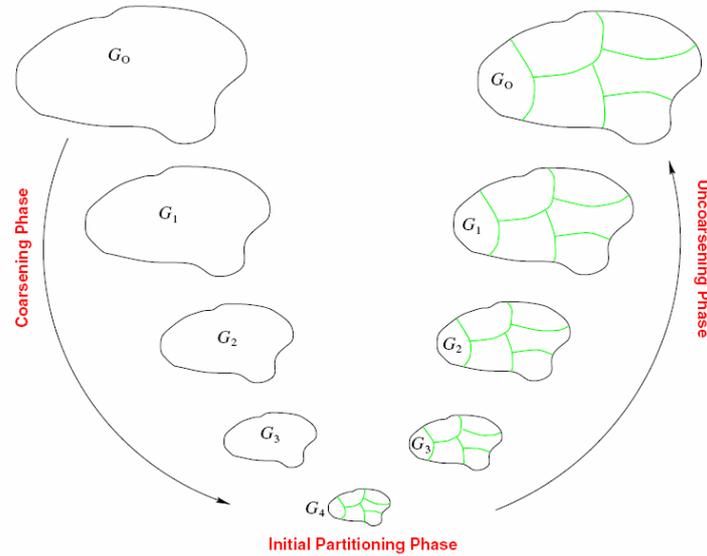


FIGURA 2.8 – Principais passos dos algoritmos de multi-níveis para o PPG descritos por Karypis (2002)

Karypis e Kumar (1998a,b) desenvolveram um aplicativo denominado METIS (<http://www.cs.umn.edu/~metis>) para o PPG, baseado nos algoritmos de multi-níveis. Experimentos realizados pelos autores (Karypis e Kumar, 1998a,c; Abou-Rjeili e Karypis, 2006) mostram que o METIS é capaz de realizar particionamento de boa qualidade em questão de segundos, mesmo em grafos com milhares de vértices.

No METIS, um grafo $G = (V, A)$ com n vértices e m arestas pode ser armazenado em um arquivo texto que deve conter $n + 1$ linhas. A primeira linha contém informações sobre o tamanho e o tipo de grafo, enquanto as demais n linhas apresentam informações para cada vértice de G .

A primeira linha contém dois (n, m) , três (n, m, fmt) ou quatro $(n, m, fmt, ncon)$ parâmetros inteiros. Os primeiros dois inteiros (n, m) indicam, respectivamente, o número de vértices e arestas. O terceiro inteiro (fmt) permite indicar se o grafo possui ou não pesos associados com os vértices, arestas ou ambos. Se $fmt = 0$, o grafo não possui pesos associados a vértices e arestas. O grafo possui pesos nas arestas se $fmt = 1$. Por outro lado, se $fmt = 10$, o grafo possui pesos associados aos vértices e por último, se $fmt = 11$ o grafo possui pesos associados aos vértices e arestas. O último parâmetro $(ncon)$ indica o número de pesos associados com cada vértice do grafo. O valor associado

à *ncon* determina se o METIS deve ou não utilizar o particionamento multi-restringido.

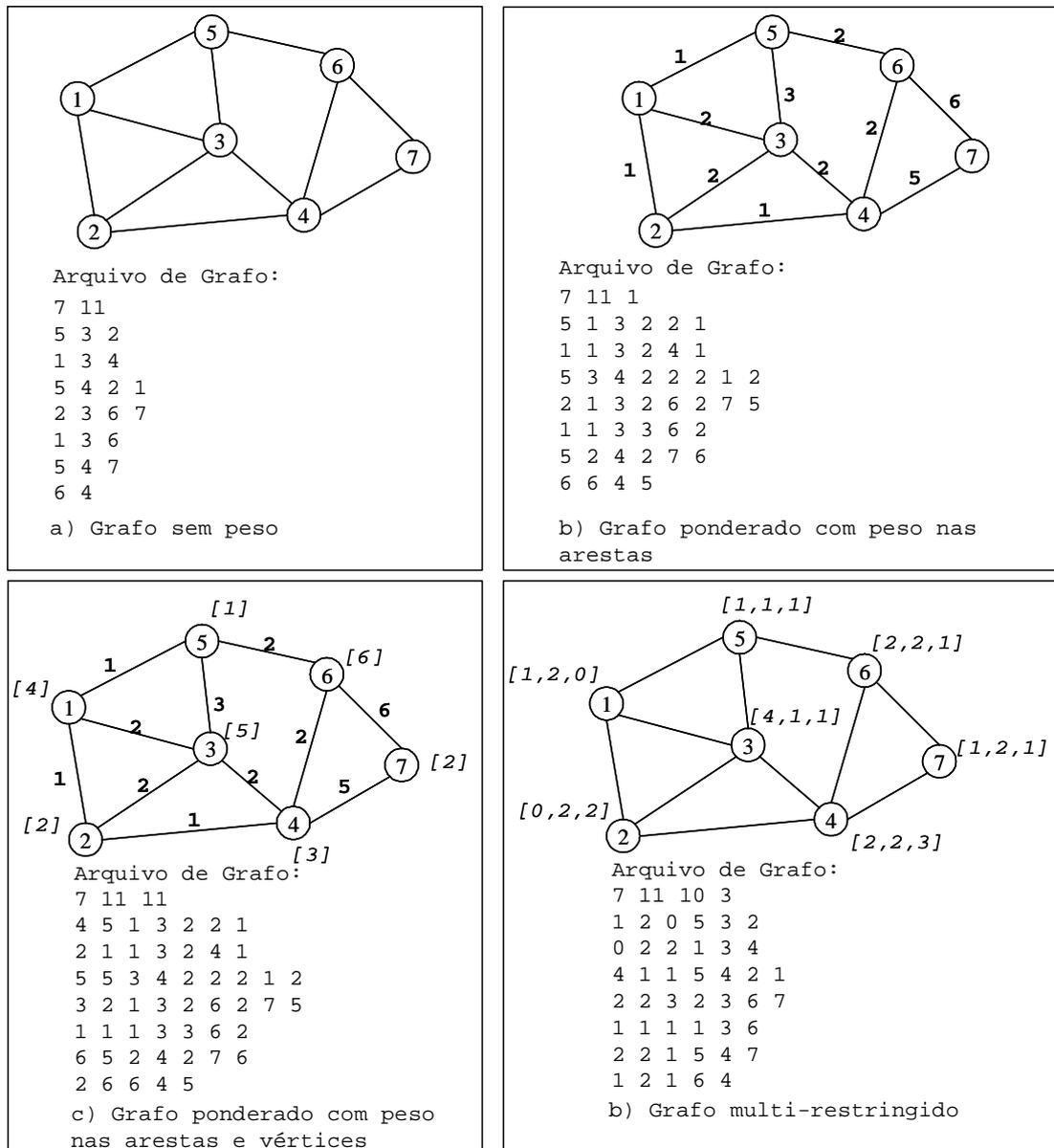


FIGURA 2.9 – Formato de arquivos no METIS para vários tipos de grafos.

Um particionamento multi-restringido é um particionamento multi-balanceado. Por exemplo, suponha que cada vértice do grafo G tenha associado dois pesos p_1 e p_2 , o objetivo do particionamento multi-restringido é encontrar \bar{P} clusters de maneira que cada um possua mais ou menos o mesmo valor total de p_1 e p_2 (Karypis, 2003).

As n linhas restantes do arquivo texto armazenam informações sobre a estrutura do grafo. A i -ésima linha contém informações relevantes do i -ésimo vértice. Dependendo do valor de fmt e $ncon$, as linhas podem ser descritas de forma diferente. No caso

mais geral, quando $fmt = 11$ e $ncon > 1$, cada linha i deve apresentar a seguinte estrutura: $w_1, w_2, \dots, w_{ncon}, v_1, a_1, v_2, a_2, \dots, v_k, a_k$, onde $w_1, w_2, \dots, w_{ncon}$ são os $ncon$ s pesos associados ao vértice i , v_1, v_2, \dots, v_k os vértices adjacentes ao vértice i , e a_1, a_2, \dots, a_k os pesos das arestas.

A Figura 2.9 apresenta quatro exemplos de arquivos (grafos) para o METIS. O formato mais simples está representado em (a). O grafo mostrado em (b) possui peso nas arestas, sendo assim, o correspondente arquivo indica tal situação através do parâmetro fmt que é igual a 1. Nesse arquivo, se o vértice i tem k vértices adjacentes, então a linha correspondente a i tem $2 \times k$ elementos. Cada par de números indica o vértice ao qual i está conectado e o peso da aresta.

A Figura 2.9 (c) apresenta um grafo com pesos nos vértices e arestas. Nesse caso o parâmetro fmt recebe o valor 11 e cada linha do arquivo armazena o peso do vértice seguido dos vértices adjacentes e dos pesos das arestas. Por último, em (d) é mostrado um grafo com múltiplos pesos nos vértices sem peso nas arestas. Nesse caso, fmt recebe o valor 10 e, como cada vértice tem três pesos associados, o parâmetro $ncon$ recebe o valor 3. Cada linha no arquivo armazena os três pesos dos vértices seguidos da lista de adjacência.

O METIS possui uma biblioteca conhecida como METISLib que pode ser chamada por programas escritos em FORTRAN ou C. Essa biblioteca contém rotinas que permitem passar um grafo como parâmetro, solicitar o particionamento e receber o resultado também como parâmetro. Essas facilidades aumentam o potencial do METIS em termos de aplicação.

Para maiores informações, veja a página *web* do METIS e o seu manual que está disponível em <http://glaros.dtc.umn.edu/gkhome/metis/metis/download>.

2.6.5 Considerações Finais

Este capítulo procurou apresentar a fundamentação teórica sobre os principais tópicos abordados neste trabalho, ou seja, relaxação lagrangeana, método de geração de colunas, teoria de planos de corte e a representação de problemas de OC através de grafos.

Como foi mostrado, o limitante de uma relaxação lagrangeana permite avaliar a qualidade de uma solução viável.

A LagClus, que será apresentada no próximo capítulo, está baseada na decomposição do problema original em subproblemas. Essa decomposição é possível a partir da

representação do problema por meio de um grafo de conflitos que é então particionado em *clusters* (subproblemas). Após esta etapa, as arestas que conectam os *clusters* são relaxadas no sentido lagrangeano.

Para aplicação da LagClus, tanto o problema original quanto os subproblemas, podem ser fortalecidos previamente com restrições de corte. Essa estratégia permite obter melhores limitantes lagrangeanos e aplicar também o algoritmo de geração de colunas, dado a equivalência entre decomposição Dantzig-Wolfe e relaxação lagrangeana.

CAPÍTULO 3

RELAXAÇÃO LAGRANGEANA COM CLUSTERS

O presente capítulo tem como objetivo apresentar a Relaxação Lagrangeana com *Clusters* (LagClus). O método é baseado no particionamento do grafo de conflitos em agrupamentos, formando subproblemas.

Na LagClus, as arestas que conectam os subproblemas são relaxadas no sentido lagrangeano, e a relaxação obtida é dividida conforme o número de subproblemas.

Com o particionamento do grafo de conflitos podem ser identificadas restrições de acoplamento e uma matriz de restrições com estrutura em blocos, permitindo, portanto, a aplicação da decomposição D-W.

Neste capítulo ainda é apresentada uma aplicação da LagClus e do AGC em uma instância pequena do PMCIV. Esses dois exemplos permitem mostrar a estrutura lógica dos dois métodos.

Discussões sobre o processo de aplicação da LagClus e do AGC também são realizadas, principalmente no que diz respeito a obtenção de bons limitantes, formas de fortalecer os subproblemas com planos de corte e a equivalência entre a LagClus e o AGC.

3.1 A LagClus e Grafos de Conflitos

A LagClus surgiu a partir da observação de que muitos grafos de conflitos relacionados aos problemas de OC apresentavam-se esparsos, bem adaptados para um estágio de particionamento de grafos. A Figura 3.1 apresenta um grafo de conflitos esparsos com dois *clusters* bem definidos (mesmo número de vértices em cada *clusters*).

Na literatura, um grafo esparsos é um grafo com “poucas” arestas, enquanto que, por outro lado, um grafo denso é um grafo no qual o número de arestas é muito próximo do número máximo de arestas (Diestel, 2000). Conforme Cormen et al. (2002) e Diestel (2000), a distinção entre um grafo esparsos e um grafo denso é um pouco vaga. Uma maneira de caracterizar um grafo $G = (V, A)$ como esparsos é escolher um k com $1 < k < 2$ tal que $|A| = O(|V|^k)$, onde $|A|$ representa o número de arestas, $|V|$ o número de vértices e a letra O refere-se a notação O de complexidade de algoritmos (Preiss, 1998) que, neste caso, indica que $|A|$ cresce, no máximo, tão rápido quanto $|V|^k$ e que $|V|^k$ domina assintoticamente $|A|$.

Voltando ao grafo mostrado na Figura 3.1, note que o número de arestas (57) é um

pouco maior do que o dobro do número de vértices (24), com isso este grafo pode ser classificado como esparso.

Suponha que o grafo de conflitos $G = (V, A)$ da Figura 3.1 esteja associado a um (PMCIIV) $\{Max\ cx : x_i + x_j \leq 1 \forall (i, j) \in A, x_i \in \{0, 1\} \forall i \in V\}$. Agora note que o único conjunto de restrições associado a este tipo de problema é composto por restrições de adjacência. Usando a relaxação lagrangeana tradicional sobre o conjunto de restrições de adjacência, tem-se:

$$v(L_\lambda PMCIIV) = Max \sum_{i=1}^{|V|} c_i x_i - \sum_{(i,j) \in A} \lambda_{i,j} (x_i + x_j) + \sum_{(i,j) \in A} \lambda_{i,j} \quad (3.1)$$

($L_\lambda PMCIIV$) Sujeito a:

$$x_i \in \{0, 1\} \forall i \in V \quad (3.2)$$

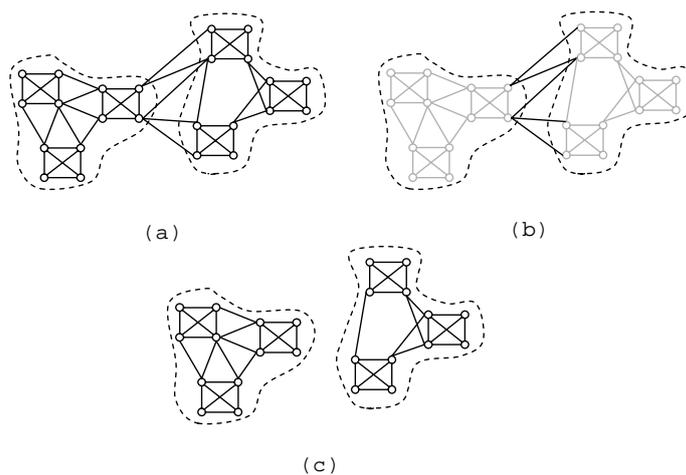


FIGURA 3.1 – Exemplo de um grafo de conflitos com regiões bem definidas: (a) grafo de conflitos, (b) arestas de conexão e (c) *clusters* independentes.

Um ponto positivo dessa relaxação é que o problema relaxado 3.1-3.2 é de fácil solução, pois, por inspeção, é possível encontrar a sua solução ótima. Porém, note que esta relaxação apresenta também pontos negativos. Repare que o problema relaxado não apresenta nenhuma restrição além da que todas as variáveis sejam binárias (restrição 3.2). Isso pode levar a um limitante superior ruim além de muitas soluções relaxadas não factíveis para o problema primal.

Por outro lado, o grafo de conflitos mostrado na Figura 3.1(a) apresenta dois *clusters* bem definidos. Assim, se as arestas que conectam os *clusters* (Figura 3.1(b)) forem relaxadas no sentido lagrangeano, dois subproblemas independentes surgem com as

mesmas características do original, ou seja, são PMCIV's também (Figura 3.1(c)).

Ao resolver os subproblemas independentemente, um limitante pode ser gerado e, como parte das restrições do problema original é considerada nos subproblemas, um limitante superior melhor ou igual ao do método tradicional pode ser obtido. Além disso, as soluções relaxadas agora são factíveis nos *clusters*, conseqüentemente, “factíveis” em partes do problema original. Comparando as soluções relaxadas do método tradicional da relaxação lagrangeana com as do método usando particionamento, o número de violações de restrições no primeiro é maior do que no segundo.

A divisão do grafo de conflitos em *clusters* e a relaxação das arestas de conexão dos *clusters* é a principal idéia da Relaxação Lagrangeana com *Clusters* (LagClus).

A LagClus surgiu a partir do trabalho de Hicks et al. (2004) no qual os autores desenvolveram um algoritmo de *Branch-and-Price* para o Problema do Máximo Conjunto Independente de Vértices com Peso (PMCIVP). Primeiro, o grafo de conflitos é particionado em *clusters* e então o problema original é modelado conforme a decomposição Dantzig-Wolfe. Cada *cluster* gera então novas colunas para o PMR. Juntamente com uma árvore de busca, os autores exploram um *Branch-and-Price* que gera bons resultados.

3.1.1 Principais Passos da LagClus

Conforme descrito na seção anterior, a LagClus é uma relaxação lagrangeana simples de ser implementada que utiliza o princípio de “Dividir para Conquistar”. Dado o grafo de conflitos $G = (V, A)$ do problema de OC, são três os passos básicos da LagClus:

- a) Aplique uma heurística de particionamento de grafos para dividir o grafo G em \bar{P} *clusters*. O problema correspondente pode ser escrito através de sua função objetivo e suas restrições de conflitos que agora estão divididas em dois grupos: um com as restrições de conflitos correspondentes às arestas internas de cada *cluster*, e um outro com as restrições de conflitos que conectam os *clusters*;
- b) Usando multiplicadores lagrangeanos λ , relaxe as restrições de conflitos que conectam os *clusters*; e
- c) Faça a decomposição do problema original em \bar{P} *clusters* e resolva a relaxação lagrangeana resultante.

Note que os subproblemas gerados neste caso possuem as mesmas características do problema original, mas de menor tamanho, conseqüentemente, *solvers* comerciais de

otimização podem ser utilizados, ou até mesmo heurísticas eficientes que, dependendo do tamanho do problema, garantem resolvê-lo de forma ótima.

Resolver os subproblemas gerados pode ser tão difícil quanto resolver o problema original. Neste caso, pode-se optar por refazer o particionamento com um número maior de *clusters*, mas deve-se avaliar a qualidade do limitante obtido.

A LagClus pode ser mais ou menos forte dependendo de \bar{P} . Quanto menor for \bar{P} , mais forte é a relaxação enquanto que, quando maior for \bar{P} , mais próximo a LagClus está da relaxação lagrangeana tradicional de todas as arestas em G , pois, no limite, quando $\bar{P} = |V|$, cada vértice se torna um *cluster* e todas as arestas em G são relaxadas.

A LagClus foi aplicada em vários problemas de OC comprovando sua eficiência. Suas aplicações envolveram o PRCP (Ribeiro e Lorena, 2006b,c), o PCPP (Ribeiro e Lorena, 2007b), PEUC (Ribeiro e Lorena, 2006d) e PPDISO. Essas aplicações serão descritas nos próximos capítulos.

A seguir, a LagClus é mostrado através de um exemplo simples que considera o PMCIV.

3.1.2 LagClus e o PMCIV

Considere um PMCIV como o da Figura 3.1 e \bar{P} o número de *clusters* obtido após a execução do primeiro passo da LagClus. Com isso, o PMCIV agora pode ser escrito como:

$$v(\text{PMCIV}) = \text{Max} \sum_{p=1}^{\bar{P}} x^p \quad (3.3)$$

(PMCIV) Sujeito a:

$$+A^1x^1 \quad +A^2x^2 \quad \dots \quad +A^{\bar{P}}x^{\bar{P}} \leq 1 \quad (3.4)$$

$$\begin{aligned} +D^1x^1 & \leq 1 \\ & +D^2x^2 & \leq 1 \\ & & \dots & \leq \vdots \\ & & & +D^{\bar{P}}x^{\bar{P}} & \leq 1 \end{aligned} \quad (3.5)$$

$$x^1 \in B^{n_1} \quad \dots \quad \dots \quad x^{\bar{P}} \in B^{n_{\bar{P}}} \quad (3.6)$$

Sendo:

- A^p uma matriz binária de dimensão $M \times |V|$ que representa os coeficientes associados às variáveis x^p do *cluster* p que aparecem nas M restrições de adjacência que conectam os *clusters*;
- D^p uma matriz binária de dimensão $|E| - M \times |V|$ que representa os coeficientes associados às variáveis x^p do *cluster* p que aparecem nas restrições de adjacência do *cluster* p ; e
- B^{n_p} um vetor de variáveis binárias com dimensão n_p associadas às variáveis x^p .

Agora executando o segundo passo da LagClus, ou seja, relaxando a restrição 3.4, obtém-se o seguinte problema relaxado, sendo $\lambda \in \mathfrak{R}_+^M$ multiplicadores lagrangeanos:

$$v(LG_\lambda PMCIV) = \text{Max} \left\{ \sum_{p=1}^{\bar{P}} (1 - A^{Tp}\lambda)x^p + \sum_{m=1}^M \lambda_m : x^p \in X^p, p = 1, \dots, \bar{P} \right\} \quad (3.7)$$

onde: $X^p = \{x^p \in B^{n_p}, D^p x^p \leq 1\}$, $p = 1, \dots, \bar{P}$.

Como cada subproblema $p \in \{1, \dots, \bar{P}\}$ é um PMCIV da forma:

$$v(PMCIV)_p = \text{Max} \{ (1 - A^{Tp}\lambda)x^p : x^p \in X^p \}, \quad (3.8)$$

e o problema relaxado 3.7 pode ser novamente escrito como (terceiro passo da LagClus):

$$v(LG_\lambda PMCIV) = \sum_{p=1}^{\bar{P}} v(PMCIV)_p + \sum_{m=1}^M \lambda_m. \quad (3.9)$$

Agora, o problema relaxado pode ser decomposto em \bar{P} subproblemas e o dual lagrangeano resolvido com um algoritmo de subgradientes.

3.1.3 Aplicando a LagClus sobre uma Instância do PMCIV

Nesta seção será apresentada uma aplicação prática da LagClus sobre uma instância do PMCIV. O exemplo apresentado é pequeno, porém permite ilustrar todos os passos e estruturas matriciais apresentados na seção anterior.

Considere a instância mostrada na Figura 3.2. O PMCIV pode ser modelado como mostrado abaixo.

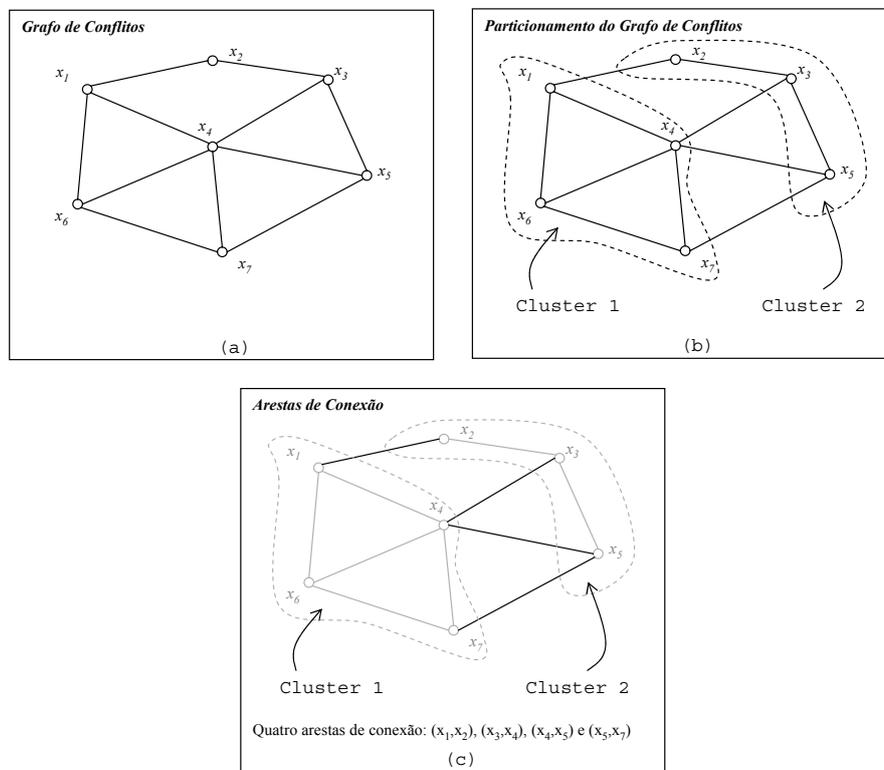


FIGURA 3.2 – Um exemplo prático da aplicação da LagClus sobre uma instância do PMCIV.

$$v(\text{PMCIV}) = \text{Max} \{x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7\} \quad (3.10)$$

(PMCIV) Sujeito a:

$$\begin{aligned}
 x_1 + x_2 &\leq 1 \\
 x_1 + x_4 &\leq 1 \\
 x_1 + x_6 &\leq 1 \\
 x_2 + x_3 &\leq 1 \\
 x_3 + x_4 &\leq 1 \\
 x_3 + x_5 &\leq 1 \\
 x_4 + x_5 &\leq 1 \\
 x_4 + x_6 &\leq 1 \\
 x_4 + x_7 &\leq 1 \\
 x_5 + x_7 &\leq 1 \\
 x_6 + x_7 &\leq 1 \\
 x_1, x_2, x_3, x_4, x_5, x_6, x_7 &\in \{0, 1\}
 \end{aligned} \quad (3.11)$$

1º Passo da LagClus

Agora suponha um particionamento do tipo 2-Particionamento como mostrado na Figura 3.2(b). As arestas mostradas em 3.2(c) são as arestas que devem ser relaxadas. Assim, este problema pode ser novamente escrito mas da seguinte maneira:

$$v(PMCIV) = \begin{array}{|c|c|} \hline \text{Cluster 1} & \text{Cluster 2} \\ \hline +x_1 & +x_2 \\ +x_4 & +x_3 \\ +x_4 & +x_5 \\ +x_7 & +x_5 \\ \hline \end{array}$$

(PMCIIV) Sujeito a:

Cluster 1	Cluster 2		
+x ₁	+x ₂	<= 1	Arestas que conectam os clusters ↑
+x ₄	+x ₃	<= 1	
+x ₄	+x ₅	<= 1	
+x ₇	+x ₅	<= 1	
+x ₁ +x ₄		<= 1	Arestas internas de cada cluster ↓
+x ₁ +x ₆		<= 1	
	+x ₂ +x ₃	<= 1	
	+x ₃ +x ₅	<= 1	
+x ₄ +x ₆		<= 1	
+x ₄ +x ₇		<= 1	
+x ₆ +x ₇		<= 1	
x ₁ , x ₄ , x ₆ , x ₇ ,	x ₂ , x ₃ , x ₅	∈ {0, 1}	

2º Passo da LagClus

O número de arestas que conectam os clusters é igual a 4 e aparecem nas primeiras quatro linhas das restrições de adjacência. Relaxando então essas restrições de conexão com multiplicadores lagrangeanos $\lambda \in \mathbb{R}_+^4$, a função objetivo do problema relaxado pode ser assim escrita:

$$v(LG_\lambda PMCIIV) = \text{Max} \left\{ \begin{array}{l} x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \\ \lambda_1(1 - x_1 - x_2) + \lambda_2(1 - x_3 - x_4) + \\ \lambda_3(1 - x_4 - x_5) + \lambda_4(1 - x_5 - x_7) \end{array} \right\}$$

que pode ser rearranjada para:

$$v(LG_\lambda PMCIV) = Max \left\{ \begin{array}{l} (1 - \lambda_1)x_1 + (1 - \lambda_2 - \lambda_3)x_4 + x_6 + (1 - \lambda_4)x_7 + \\ (1 - \lambda_1)x_2 + (1 - \lambda_2)x_3 + (1 - \lambda_3 - \lambda_4)x_5 + \\ (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) \end{array} \right\}. \quad (3.12)$$

Assim, o problema relaxado pode ser apresentado como:

$$v(LG_\lambda PMCIV) = Max \left\{ \begin{array}{l} (1 - \lambda_1)x_1 + (1 - \lambda_2 - \lambda_3)x_4 + x_6 + (1 - \lambda_4)x_7 + \\ (1 - \lambda_1)x_2 + (1 - \lambda_2)x_3 + (1 - \lambda_3 - \lambda_4)x_5 + \\ (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) \end{array} \right\} \quad (3.13)$$

(LG_λPMCIV) Sujeito a:

$$\begin{aligned} x_1 + x_4 &\leq 1 \\ x_1 + x_6 &\leq 1 \\ x_2 + x_3 &\leq 1 \\ x_3 + x_5 &\leq 1 \\ x_4 + x_6 &\leq 1 \\ x_4 + x_7 &\leq 1 \\ x_6 + x_7 &\leq 1 \\ x_1, x_4, x_6, x_7, x_2, x_3, x_5 &\in \{0, 1\} \end{aligned} \quad (3.14)$$

3º Passo da LagClus

Agora, o problema relaxado pode ser dividido nos seguintes subproblemas:

$$v(LG_\lambda PMCIV)_1 = Max \{(1 - \lambda_1)x_1 + (1 - \lambda_2 - \lambda_3)x_4 + x_6 + (1 - \lambda_4)x_7\} \quad (3.15)$$

(LG_λPMCIV)₁ Sujeito a:

$$\begin{aligned} x_1 + x_4 &\leq 1 \\ x_1 + x_6 &\leq 1 \\ x_4 + x_6 &\leq 1 \\ x_4 + x_7 &\leq 1 \\ x_6 + x_7 &\leq 1 \\ x_1, x_4, x_6, x_7 &\in \{0, 1\} \end{aligned} \quad (3.16)$$

e

$$v(LG_{\lambda}PM CIV)_2 = \text{Max} \{(1 - \lambda_1) x_2 + (1 - \lambda_2) x_3 + (1 - \lambda_3 - \lambda_4) x_5\} \quad (3.17)$$

$(LG_{\lambda}PM CIV)_2$ Sujeito a:

$$\begin{aligned} x_2 + x_3 &\leq 1 \\ x_3 + x_5 &\leq 1 \\ x_2, x_3, x_5 &\in \{0, 1\} \end{aligned} \quad (3.18)$$

Note que parte das restrições de adjacência do problema original aparece nos subproblemas.

Passo 0: Inicialização. Faça $\lambda^1 \geq 0$, $k \leftarrow 1$, $v^k \leftarrow +\infty$

Passo 1: LagClus.

$$L_{\lambda}P = \sum_{m=1}^M \lambda_m^k$$

Para p de 1 a P **Faça**

$$L_{\lambda}P = L_{\lambda}P + v(L_{\lambda}P^p)$$

Fim Para

Se $(d - Dx^k) \geq 0$ e $\lambda(d - Dx^k) = 0$,
pare pois x^k resolve o problema primal
 P e $v(L_{\lambda}P) = v(DL) = v(P)$

Passo 2: Guardar melhor solução.

Se $v^k > v(L_{\lambda}P) = cx^k + \lambda^k(d - Dx^k)$, faça $v^k \leftarrow v(L_{\lambda}P)$. Caso contrário faça $v^k \leftarrow v^{k-1}$.

Passo 3: Calcular passo do subgradiente.

Ache um novo ponto
 $\lambda^{k+1} = \lambda^k + \pi_k(d - Dx^k) / \|d - Dx^k\|$, sendo π_k o
tamanho do passo que satisfaz a
seguinte condição:

- 1) $\pi_k > 0$ para todo k ;
- 2) $\lim_{k \rightarrow \infty} \pi_k = 0$
- 3) $\sum_{k=1}^{\infty} \pi_k = +\infty$

Passo 4: Projeção.

Projete o novo valor λ^{k+1} fazendo:

$$\lambda_j^{k+1} \leftarrow \text{Max}\{0, \lambda_j^{k+1}\} \forall j.$$

Faça $k \leftarrow k+1$ e volte ao passo 1.

FIGURA 3.3 – Adaptação do algoritmo de subgradientes para contemplar a LagClus.

Portanto, o problema relaxado 3.13-3.14 pode ser escrito como:

$$v(LG_\lambda PMCIV) = \sum_{p=1}^2 v(LG_\lambda PMCIV)_p + (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) \quad (3.19)$$

O Passo 1 do algoritmo de subgradientes descrito na Seção 2.2.2 deve ser alterado conforme mostra a Figura 3.3 para contemplar a decomposição do problema original em *clusters*, como indicado na equação 3.19.

3.1.4 Aspectos da LagClus que devem ser Observados

Como observado anteriormente, a LagClus é dependente do número de arestas relaxadas, ou seja, quanto maior é esse número, mais fraca fica a LagClus. As arestas relaxadas estão diretamente associadas ao algoritmo de particionamento e ao número de *clusters* considerado. Isso porque o algoritmo de particionamento pode não ser eficiente e conseqüentemente muitas arestas podem ser desconsideradas. E, por outro lado, quanto maior o número de *clusters*, maior o número de arestas particionadas.

Como visto no Capítulo 2, o problema de particionamento de grafos é *NP-Hard*, portanto heurísticas eficientes podem ser utilizados na LagClus, como é o caso da heurística METIS de Karypis e Kumar (1998a). Além disso, dependendo do problema, o particionamento pode e deve ser feito considerando pesos nas arestas e/ou nos vértices, e novamente o METIS pode ser aplicado.

Por outro lado, estratégias de fortalecimento de formulações que envolvem restrições de corte podem ser inseridas nos subproblemas. No caso do exemplo mostrado na Seção 3.1.3, se o vértice x_1 apresentasse uma restrição de adjacência em relação ao vértice x_7 , todas as restrições apresentadas no *cluster* 1 poderiam ser substituídas por uma clique envolvendo os vértices x_1, x_4, x_6 e x_7 .

3.2 Algoritmo de Geração de Colunas

Tomando o PMCIV como exemplo, a decomposição mostrada em 3.3-3.6 pode ser explorada em um Algoritmo de Geração de Colunas, conforme mostrado no Capítulo 2. Sendo assim, aplicando a decomposição Dantzig-Wolfe (DW) sobre a relaxação de programação linear (PL) do problema 3.3-3.6, o seguinte PMR é obtido:

$$v(PMCIV_{DW})_{LP} = Max \left\{ \sum_{p=1}^{\bar{P}} \sum_{j \in J_p} w^{j,p} \lambda_{j,p} \right\} \quad (3.20)$$

($PMCIV_{DW}$) $_{PL}$ Sujeito a:

$$\sum_{p=1}^{\bar{P}} \sum_{j \in J_p} \lambda_{j,p} [A_p] x^{j,p} \leq 1 \quad (3.21)$$

$$\sum_{j \in J_p} \lambda_{j,p} = 1 \forall p \in \{1, \dots, \bar{P}\} \quad (3.22)$$

$$\lambda_{j,p} \geq 0 \forall p \in \{1, \dots, \bar{P}\} \forall j \in J_p \quad (3.23)$$

Sendo:

- $\lambda_{j,p}$ uma variável de decisão que representa o ponto extremo $j \in J_p$;
- J_p o conjunto de pontos extremos do *cluster* p ; e
- $w^{j,p}$ o custo da solução x^j do subproblema p .

Os \bar{P} subproblemas considerados aqui são os mesmos mostrados em 3.8, mas com as variáveis duais (Δ) correspondentes às restrições descritas em 3.21:

$$v(PMCIV)_p = Max \{ (1 - A^{Tp} \Delta) x^p : x^p \in X^p \} \quad (3.24)$$

Nesse caso, uma nova coluna fornecida pelo p -ésimo *cluster* é uma coluna de melhora se $v(PMCIV)_p - \beta_p > 0$, sendo β_p a variável dual associada à p -ésima restrição de convexidade descrita em 3.22.

Note que a LagClus definida pela equação 3.9 também pode ser obtida diretamente do PMR descrito acima pelo modelo 3.20-3.23 usando a seguinte formulação:

$$v(LG_{\Delta}PMCIV) = \sum_{p=1}^{\bar{P}} v(PMCIV)_p + \sum_{m=1}^M \Delta_m. \quad (3.25)$$

A relaxação descrita em 3.25 mostra a equivalência entre o dual lagrangeano e a decomposição de Dantzig-Wolfe. Porém, note que no AGC um número inicial de colunas devem ser inseridas no PMR. Essa característica pode ser útil principalmente se existe uma boa heurística primal para o problema considerado. Questões de convergência podem ser gerenciadas usando um bom conjunto inicial de colunas e estratégias de eliminação de colunas improdutivas.

3.2.1 Aplicando o AGC sobre uma Instância do PMCIV

Considere novamente o PMCIV mostrado na Figura 3.2. Seja x_1 e x_2 duas soluções viáveis para o problema tal que $x_1 = (1, 0, 1, 0, 0, 0, 1)$ e $x_2 = (0, 1, 0, 1, 0, 0, 0)$. Essas duas soluções podem ser decompostas nos dois *clusters*: $x^{1,1} = (1, 0, 0, 0, 0, 0, 1)$, $x^{1,2} = (0, 0, 1, 0, 0, 0, 0)$, $x^{2,1} = (0, 0, 0, 1, 0, 0, 0)$ e $x^{2,2} = (0, 1, 0, 0, 0, 0, 0)$. Isso significa que inicialmente existem dois pontos extremos para cada *cluster*, ou seja, $J_1 = 2$ e $J_2 = 2$, representados da seguinte maneira, $x^{1,1}$ por $\lambda^{1,1}$, $x^{1,2}$ por $\lambda^{1,2}$, $x^{2,1}$ por $\lambda^{2,1}$ e $x^{2,2}$ por $\lambda^{2,2}$.

Considerando a função objetivo do PMCIV como descrito em 3.3, as soluções viáveis x_1 e x_2 representam os valores 3 e 2, respectivamente, para $v(\text{PMCIV})$. Agora analisando os *clusters*, os valores dessas soluções podem ser assim divididos: $w^{1,1} = 2$, $w^{1,2} = 1$, $w^{2,1} = 1$ e $w^{2,2} = 1$. Sendo assim, cada variável $x^{i,p}$ descrita acima, corresponde às

seguintes colunas no PMR: $x^{1,1} \leftrightarrow \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$, $x^{1,2} \leftrightarrow \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$, $x^{2,1} \leftrightarrow \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ e $x^{2,2} \leftrightarrow \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$.

Lembre-se que, conforme a Seção 2.3.1, no topo de cada coluna do PMR está o custo da coluna.

Sendo assim, o PMR do PMCIV mostrado na Figura 3.2 pode ser assim descrito:

$$v(\text{PMCIV}_{DW})_{LP} = \text{Max} \{2\lambda_{1,1} + 1\lambda_{1,2} + 1\lambda_{2,1} + 1\lambda_{2,2}\} \quad (3.26)$$

(PMCI V_{DW}) $_{LP}$ Sujeito a:

$$\begin{array}{rcl}
 +\lambda_{1,1} & & +\lambda_{2,2} \leq 1 \\
 & +\lambda_{2,1} & +\lambda_{1,2} \leq 1 \\
 & +\lambda_{2,1} & \leq 1 \\
 +\lambda_{1,1} & & \leq 1
 \end{array} \tag{3.27}$$

$$\begin{array}{rcl}
 +\lambda_{1,1} & +\lambda_{2,1} & = 1 \\
 & +\lambda_{1,2} & +\lambda_{2,2} = 1
 \end{array} \tag{3.28}$$

$$\lambda_{1,1}, \lambda_{2,1}, \lambda_{1,2}, \lambda_{2,2} \geq 0 \tag{3.29}$$

Repare que, como as soluções x_1 e x_2 são soluções viáveis para o problema, não existem violações das restrições entre *clusters* (arestas de conexão). Por outro lado, as colunas $\lambda_{1,1}$ e $\lambda_{2,2}$ não podem compor a solução do problema, pois caso contrário, a restrição $x_1 + x_2 \leq 1$ será violada. A mesma coisa acontece com $\lambda_{2,1}$ e $\lambda_{1,2}$ mas nesse caso, a restrição $x_3 + x_4 \leq 1$ que estará violada.

Agora suponha que os subproblemas gerem as seguintes soluções: $x^{3,1} = (0, 0, 0, 1, 0, 1, 0)$ e $x^{3,2} = (0, 1, 0, 0, 1, 0, 0)$. Neste caso, essas duas soluções representam a solução $x^3 = (0, 1, 0, 1, 1, 1, 0)$ que não é viável para o PMCI V , pois viola a restrição $x_4 + x_5 \leq 1$. Mas note que as colunas $x^{3,1}$ e $x^{3,2}$ formam uma solução viável. Esse comportamento é interessante, uma vez que durante o processo de geração de colunas, uma nova coluna de um *cluster* combinada com uma coluna previamente obtida de outro *cluster*, pode gerar uma solução viável melhor do que todas as obtidas até o momento.

Para $x^{3,1}$ e $x^{3,2}$, sejam $v(\text{PMCI}V)_1 - \beta_1 > 0$ e $v(\text{PMCI}V)_2 - \beta_2 > 0$ expressões verdadeiras tal que $w^{3,1} = 2$ e $w^{3,2} = 2$. Com isso, as seguintes colunas devem ser

$$\text{inseridas: } x^{3,1} \leftrightarrow \begin{bmatrix} 2 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \text{ e } x^{3,2} \leftrightarrow \begin{bmatrix} 2 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

O PMR agora pode ser assim descrito:

$$v(\text{PMCIV}_{DW})_{LP} = \text{Max} \{2\lambda_{1,1} + 1\lambda_{1,2} + 1\lambda_{2,1} + 1\lambda_{2,2} + 2\lambda_{3,1} + 2\lambda_{3,2}\} \quad (3.30)$$

($\text{PMCIV}_{DW})_{LP}$ Sujeito a:

$$\begin{array}{rcccccl} +\lambda_{1,1} & & & +\lambda_{2,2} & & +\lambda_{3,2} & \leq & 1 \\ & +\lambda_{2,1} & +\lambda_{1,2} & & +\lambda_{3,1} & & \leq & 1 \\ & +\lambda_{2,1} & & & +\lambda_{3,1} & +\lambda_{3,2} & \leq & 1 \\ +\lambda_{1,1} & & & & & +\lambda_{3,2} & \leq & 1 \end{array} \quad (3.31)$$

$$\begin{array}{rcccccl} +\lambda_{1,1} & +\lambda_{2,1} & & & +\lambda_{3,1} & & = & 1 \\ & & +\lambda_{1,2} & +\lambda_{2,2} & & +\lambda_{3,2} & = & 1 \end{array} \quad (3.32)$$

$$\lambda_{1,1}, \lambda_{2,1}, \lambda_{1,2}, \lambda_{2,2}, \lambda_{3,1}, \lambda_{3,2} \geq 0 \quad (3.33)$$

No novo PMR obtido, repare nas restrições descritas em 3.31. Note que as incompatibilidades entre as soluções dos *clusters* são atualizadas explicitamente na medida em que novas colunas são introduzidas no PMR, bem como as restrições de que uma coluna de cada *cluster* deve ser selecionada (veja restrição 3.32).

O exemplo mostrado permite avaliar e entender o funcionamento do AGC, principalmente no que diz respeito ao processo de inserção de novas colunas. Os passos aqui executados foram baseados no AGC básico descrito na Figura 2.2.

3.2.2 Aspectos que devem ser Observados no AGC que usa a Idéia dos *Clusters*

A técnica de Geração de Colunas pode apresentar problemas de convergência, gerando muitas colunas que tornam o PMR difícil de ser resolvido. Para contornar isso, vários autores propõem estratégias para eliminar colunas improdutivas (veja, por exemplo, Senne et al. (2005) e Senne et al. (2007)).

Mas no caso da AGC que usa a idéia dos *clusters*, essas estratégias devem ser estudadas de maneira a evitar inconsistências, pois cada *cluster* p deve obrigatoriamente ter uma coluna no PMR (veja restrição 3.22) e pelo menos uma solução viável para o PMCIV original deve existir, ou seja, ao selecionar uma coluna de cada *cluster*, a solução gerada não deve violar as restrições 3.21.

A aplicação direta do método de geração de colunas freqüentemente produz um número

muito grande colunas que não são relevantes para a solução final, comprometendo assim a convergência para a solução do problema e produzindo os chamados problemas de estabilização (Lorena et al., 2003). Nestes casos, observa-se que as variáveis duais oscilam em torno da solução dual ótima, logo métodos que previnam tal comportamento podem acelerar a resolução do problema. Dentre estes merecem destaque: o uso da relaxação Lagrangeana/*Surrogate* (LagSur), proposta por Narciso e Lorena (1999), no qual o multiplicador LagSur modifica o critério de custo reduzido, fornecendo melhores colunas para o AGC (veja Senne et al. (2005); Pereira (2005); Senne et al. (2007)); Método *Boxstep* (Marsten et al., 1975), que restringe o espaço de busca de soluções duais a uma região limitada tendo a solução dual atual como centro; Método *Analytic Centre Cutting Plane* (du Merle et al., 1998), que usa o centro analítico de uma região da função dual como solução, no lugar da solução ótima, não permitindo assim mudanças muito drásticas entre as soluções duais de duas iterações consecutivas; Métodos *Bundle* (Briant et al., 2005), que combinam regiões de confiança e penalizações para que as soluções duais não variem muito de uma iteração para outra.

Para maiores informações sobre outros algoritmos e os detalhes dos métodos descritos anteriormente, ver Neame (1999); Senne et al. (2005); Pereira (2005); Senne et al. (2007).

Passo 0: Inicialização.
 Determine um conjunto viável inicial de colunas para o PMR.
 Faça $i \leftarrow 1$

Passo 1: Resolva o PMR_{PL} .

Passo 2: Resolva o subproblema gerador de colunas.
 Com os valores duais (π, μ) do PMR_{PL} obtidos no Passo 2, atualize e resolva o subproblema.

Passo 3: Adicione colunas no PMR_{PL} .
 Adicione as novas colunas com custo reduzido positivo no PMR_{RL} , faça $i \leftarrow i+1$ e volte ao Passo 1.

Se não existirem novas colunas ou $v(PMR)_{PL} = v(LG_{\lambda}P)$, pare pois a solução é ótima.

FIGURA 3.4 – AGC alterado para contemplar o limitante da LagClus.

Outro ponto interessante a ser observado em um AGC com a idéia de *clusters* é a estratégia para gerar as soluções iniciais viáveis que deverão compor o primeiro conjunto de colunas. Experimentos computacionais, que serão apresentados nos próximos capítulos, mostraram que mesmo soluções iniciais não tão boas devem ser levadas em consideração, pois, ao particionar essas soluções nos respectivos *clusters*, a solução de um

cluster pode ser interessante de tal maneira que ao combiná-la com as soluções possíveis dos demais *clusters*, uma solução de boa qualidade pode ser obtida. Esse comportamento foi apresentado no exemplo da Seção 3.2.1.

Dada a relação entre LagClus e AGC, conforme mostrada pela equação 3.25, é possível utilizar o limitante gerado pela LagClus para parar o processo de geração de colunas. Experimentos computacionais realizados com o PCPP mostraram que o limitante da PMR encontra o limitante da LagClus, mostrando que o processo pode ser finalizado sem gerar todas as colunas. Conseqüentemente, o passo 3 do AGC descrito na Seção 2.3.2 pode ser alterado para contemplar esse critério de parada, conforme mostra a Figura 3.4.

3.3 Considerações Finais do Capítulo

Este capítulo apresentou a LagClus e um AGC para problemas modelados em grafos de conflitos. A idéia consiste em dividir o grafo de conflitos em *clusters* e na LagClus, relaxar as arestas que conectam os *clusters* enquanto que no AGC, essas arestas de conexão são utilizados no PMR.

Uma instância pequena do PMCIIV foi utilizada para exemplificar a aplicação da LagClus e do AGC. Na LagClus os subproblemas são resolvidos gerando uma solução relaxada, mas como parte das restrições do problema original são consideradas nos *clusters*, essa solução relaxada está parcialmente não violada, indicando que uma heurística lagrangeana pode ser utilizada para compor uma solução viável para um problema.

Questões como o fortalecimento da LagClus foram levantadas. Basicamente, o fortalecimento da LagClus pode ser obtido considerando um número menor de *clusters* e a inserção de cortes nos subproblemas.

Por outro lado, os critérios de convergência do AGC foram discutidos e indicaram que, dependendo do problema, heurísticas primais que geram soluções não muito boas são interessantes para a AGC, porque quando estas são divididas nos *clusters*, podem gerar soluções boas para cada *cluster* que combinadas, aceleram o processo de convergência. Além disso, a questão da remoção de colunas também foi explorada, indicando os problemas que podem ocorrer se não for considerada a estrutura lógica embutida na decomposição Dantzig-Wolfe aqui apresentada.

Com isso, dado a equivalência entre LagClus e AGC mostrada pela equação 3.25, o limitante da LagClus pode ser utilizado para parar o processo de geração de colunas, podendo reduzir assim, o tempo computacional.

Nos próximos capítulos serão apresentados resultados computacionais da LagClus e AGC aplicados em problemas de otimização modelados sobre grafo de conflitos. Os resultados mostraram que tanto a LagClus quanto o AGC são métodos fortes capazes de fechar *gaps* de dualidade em um tempo computacional reduzido.

sobreposições sejam inevitáveis. Com isso, existem na literatura duas outras abordagens para esse problema: a primeira considera o Problema do Máximo Número de Rótulos Sem Conflitos (PMNRSC) e a segunda o Problema da Minimização do Número de Conflitos (PMNC).

Não se conhece na literatura até o presente momento, uma formulação matemática de programação linear inteira binária para o PMNRSC, já o PMNC possui duas formulações: uma baseada nas arestas do grafo de conflitos e outra baseada nos pontos a serem rotulados. Essas duas formulações foram propostas por Ribeiro e Lorena (2004a,b).

O PMNRSC procura maximizar o número de pontos rotulados sem conflitos não levando em consideração o aspecto visual do mapa, podendo apresentar áreas ilegíveis. Por outro lado, o PMNC procura rotular todos os pontos, minimizando os conflitos gerados. Essa última abordagem é interessante, pois, se em um mapa os conflitos são inevitáveis, esses poderão ser “espalhados” pelo mesmo, reduzindo assim a ocorrência de áreas ilegíveis.

4.1 Padronização Cartográfica e o Grafo de Conflitos do PRCP

Pela Figura 4.1 pode-se observar nas áreas indicadas pelas setas, os conflitos entre os rótulos. Uma maneira de resolver esse problema seria movimentar os rótulos, afastando-os uns dos outros. No entanto, os rótulos devem estar próximos dos pontos obedecendo a uma padronização cartográfica, o que limita as suas possíveis posições.

A padronização cartográfica mais conhecida é a de Christensen et al. (1995). Ao se definir essas possíveis localizações do rótulo de cada ponto, este problema pode ser formulado como um problema de OC. A Figura 4.2 mostra um conjunto de 8 possíveis posições para o rótulo de um ponto, conhecidas por posições candidatas. O número presente em cada uma determina a preferência cartográfica, sendo a posição 1 a de maior interesse para a cartografia.

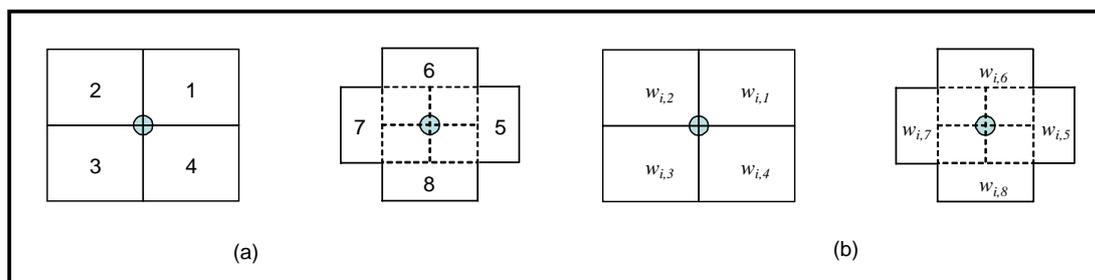


FIGURA 4.2 – Padronização cartográfica proposta por Christensen et al. (1995).

Considere um problema em que para cada ponto existem somente 4 posições candidatas,

como mostra a Figura 4.3. O mesmo pode ser facilmente representado por um grafo de conflitos. Sendo NP o número de pontos a serem rotulados e PC o número de posições candidatas de cada ponto, seja $G = (V, A)$ o grafo de conflitos tal que $V = \{v_1, v_2, \dots, v_{NP \times PC}\}$ representa o conjunto das posições candidatas (vértices) e $A = \{(v_i, v_j) : i, j \in V, i \neq j\}$ as possíveis sobreposições (conflitos). A Figura 4.3(b) apresenta o grafo de conflitos obtido para o problema da Figura 4.3(a), e na Figura 4.3(c) a solução ótima para esse problema. Para se determinar a qualidade de uma solução, utiliza-se a proporção de rótulos livres (sem conflitos). No caso da Figura 4.3(c) têm-se 100% de rótulos livres.

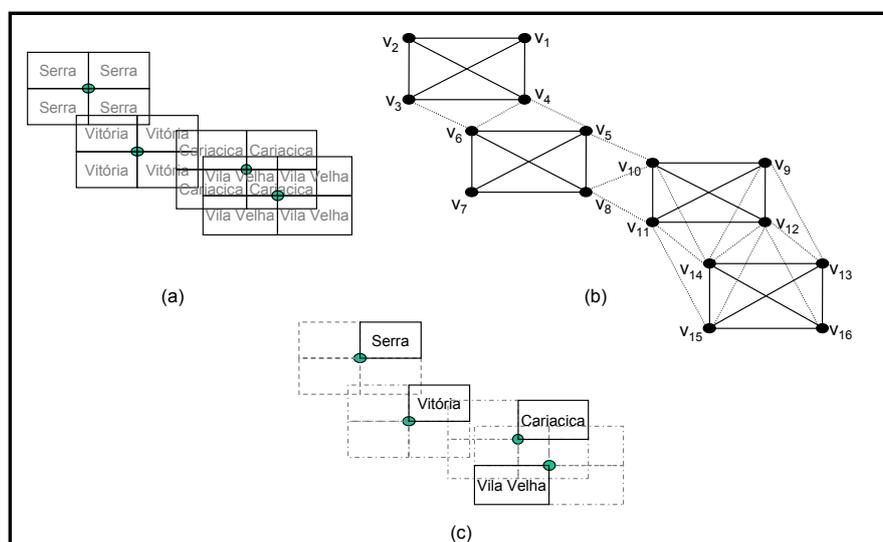


FIGURA 4.3 – Exemplo de um grafo de conflitos para o PRCP: (a) Posições candidatas, (b) grafo de conflitos e (c) solução ótima.

4.2 Revisão Bibliográfica para o PRCP

A primeira abordagem consiste em tratar o PRCP como um PMCIV, desta forma, vários trabalhos propuseram diferentes algoritmos, assim como técnicas que permitem reduzir o número de restrições. Zoraster (1986, 1990, 1991) formulou matematicamente o PRCP, porém, para lidar com as restrições de conflito, o autor criou posições candidatas fictícias de custo elevado, de tal modo que se nenhuma das reais posições candidatas pudessem ser utilizadas para posicionar um rótulo, a posição fictícia era então utilizada. Zoraster (1990) apresentou uma relaxação lagrangeana tradicional sobre o conjunto de restrições de adjacência. Essa relaxação foi aplicada em instâncias pequenas produzindo bons resultados.

Por outro lado, Strijk et al. (2000) propuseram uma modelagem de programação inteira binária que usa restrições de corte para reduzir o número de restrições de conflito

presentes no modelo de Zoraster (1990, 1991). Essas técnicas de redução apareceram antes nos trabalhos de Moon e Chaudhry (1984) e Murray e Church (1997b), e consistem em utilizar conceitos de cliques, reduzindo assim o número total de restrições do problema original. Strijk et al. (2000) utilizaram uma relaxação de programação linear e propuseram um algoritmo *Branch-and-Bound* para encontrar as soluções ótimas para os problemas testados. Devido às dificuldades encontradas, os autores ainda aplicaram e propuseram várias heurísticas, sendo elas: *Simulated Annealing*, Busca em Vizinhança Diversificada, *k*-Opt e Busca Tabu. Sendo a Busca Tabu o algoritmo que apresentou os melhores resultados.

Considerando a segunda abordagem, ou seja, o PRCP como um PMNRSC, Christensen et al. (1993, 1995) propuseram um método denominado Busca Exaustiva que faz uma procura por soluções melhores alternando as posições de rótulos previamente posicionados. Christensen et al. (1995) também propuseram um Algoritmo Guloso com sucessivas otimizações locais e um algoritmo denominado *Discrete Gradient Descent* que considera as posições alternativas dos rótulos, porém, esse algoritmo, apesar de rápido, tem dificuldades para escapar de máximos locais.

Verner et al. (1997) aplicaram um Algoritmo Genético com máscara no PRCP. Eles propuseram uma maneira de trabalhar com as máscaras de modo que, se um rótulo está em conflito, será permitido a troca de posições através de cruzamentos e mutações. Mais tarde, Yamamoto et al. (2002) propuseram um algoritmo de Busca Tabu eficiente que forneceu resultados muito bons quando comparados com a literatura.

Schreyer e Raidl (2004) utilizaram um Sistema de Colônia de Formigas, mas seus resultados não foram satisfatórios quando comparados com os obtidos por Yamamoto et al. (2002). Yamamoto e Lorena (2005) desenvolveram um algoritmo exato para o PMNRSC e aplicaram o Algoritmo Genético Construtivo proposto por Lorena e Furtado (2001). O algoritmo exato foi aplicado a instâncias de até 25 pontos, pois utiliza uma estrutura em árvore de busca, o que limitou a sua aplicação. Em contrapartida, o Algoritmo Genético Construtivo foi aplicado a instâncias com até 1000 pontos, fornecendo bons resultados quando comparados com os da Busca Tabu de Yamamoto et al. (2002).

Alvim e Taillard (2005) aplicaram a metaheurística POPMUSIC (*Partial Optimization Metaheuristic under Special Intensification Conditions*) proposta por Taillard e Voss (2001), nas instâncias propostas por Yamamoto et al. (2002). A idéia básica do POPMUSIC é otimizar, localmente, partes da solução, uma vez que uma solução viável está disponível. Essas otimizações locais são então repetidas até que um mínimo ou

máximo local seja encontrado.

Para o PRCP, Alvim e Taillard (2005) usaram a Busca Tabu proposta por Yamamoto et al. (2002) como método de busca local. Os resultados obtidos foram melhores que os propostos por Yamamoto e Lorena (2005).

A terceira abordagem que considera o PRCP como um PMNC, foi introduzida por Ribeiro e Lorena (2004a,b). O PMNC considera a minimização do número de conflitos e, em alguns casos, as soluções encontradas são limitantes inferiores para a segunda abordagem (PRCP como um PMNRSC). Considerando a Figura 4.4, as soluções mostradas em (b) e (c), sob o ponto de vista da abordagem dado pelo PMNRSC, são iguais, pois apresentam o mesmo número de rótulos em conflitos (4). Porém se considerarmos a representação em grafo mostrada na figura, a solução mostrada em (c) é mais interessante que a mostrada em (b), pois apresenta um número menor de arestas no grafo de conflitos remanescente. O PMNC, no caso de conflitos inevitáveis, permite “espalhar” mais os rótulos, fornecendo uma melhor visualização do mapa como pode ser visto na Figura 4.4(c).

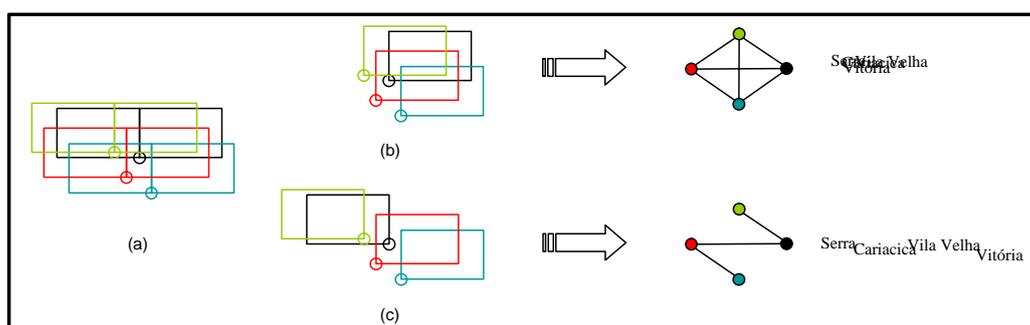


FIGURA 4.4 – Comparação entre o PMNRSC e PMNC.

Como mencionado anteriormente, existem duas modelagens para o PMNC que diferem, basicamente, no número de restrições de conflitos (adjacências) geradas. A segunda trabalha com pontos e foi inspirada no trabalho de Murray e Church (1997b). Este modelo reduz consideravelmente o número de restrições quando comparado com o primeiro baseado em arestas.

Recentemente, Cravo et al. (2007) propuseram um GRASP (*Greedy Randomized Adaptive Search Procedure*) para o PRCP. Na fase construtiva, uma solução é construída ponto a ponto, até que uma solução viável seja obtida. Na segunda fase, são executadas trocas nas posições candidatas, analisando as soluções geradas sendo que a melhor é armazenada. Entretanto, os autores mesclaram os conceitos do PMNC com os do PMNRSC. Os

resultados encontrados foram bem interessantes para as instâncias de Yamamoto et al. (2002), superando os obtidos por Yamamoto e Lorena (2005) e Alvim e Taillard (2005).

O PRCP pode gerar grafos com um alto número de arestas, dificultando o processo de otimização por qualquer uma das três abordagens. Tentando reduzir o grafo de conflitos gerado por um PRCP, Wagner et al. (2001) apresentaram um método de redução. Os autores propuseram três regras que permitem reduzir a grafo de conflitos sem alterar a solução ótima do problema. Essas regras podem ser assim descritas:

- Se p tem uma posição candidata p_i sem qualquer conflito, adicione p_i como parte da solução e elimine todas as outras posições candidatas do ponto p (veja Figura 4.5(a));
- Se p tem uma posição candidata p_i que está em conflito somente com uma posição candidata q_k , e q tem uma posição candidata q_j (com $j \neq k$) que está sobreposta somente por p_i (com $l \neq i$), então adicione p_i e q_j como parte da solução e elimine todas as outras posições candidatas de p e q (veja Figura 4.5(b)); e
- Se p tem somente uma posição candidata p_i restante, e as posições candidatas sobrepondo p_i formam uma clique, então adicione p_i como parte da solução e elimine todas as posições candidatas que conflitam com p_i (veja Figura 4.5(c)).

Todas essas regras devem ser aplicadas exaustivamente. Após eliminar uma posição candidata p_i qualquer, o procedimento de redução deve ser aplicado recursivamente na vizinhança de p_i .

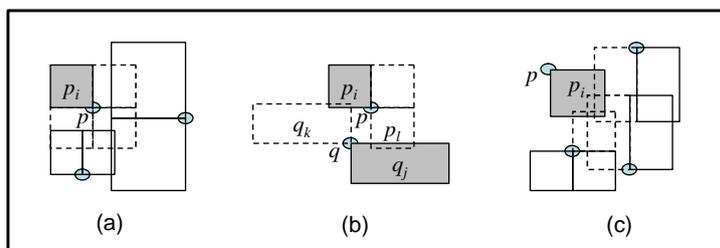


FIGURA 4.5 – Regras para reduzir o grafo de conflitos propostas por Wagner et al. (2001)

A seguir são apresentadas as duas modelagens matemáticas propostas para o PMNC.

4.3 Modelagem Matemática Baseada em Arestas - PMNC^a

A modelagem matemática baseada em arestas tem como objetivo minimizar o número de conflitos. Para cada ponto i existe um número fixo P_i de posições candidatas. Cada posição candidata é representada por uma variável binária $x_{i,j}$ tal que $i = 1, \dots, N$ e $j = 1, \dots, P_i$, sendo N o número de pontos a serem rotulados. Se $x_{i,j} = 1$ a posição candidata j do ponto i será alocada (receberá o rótulo do ponto i), caso contrário, $x_{i,j} = 0$. Além disso, a cada posição candidata é associado um custo (uma preferência cartográfica) representado por $w_{i,j}$, conforme descrito na Figura 4.2(b).

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

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

$$v(\text{PMNC}^a) = \text{Min} \left\{ \sum_{i=1}^N \sum_{j=1}^{P_i} \left(w_{i,j} x_{i,j} + \sum_{(k,t) \in S_{i,j}} y_{i,j,k,t} \right) \right\} \quad (4.1)$$

Porém, para cada ponto i somente uma posição candidata deve receber o rótulo do ponto. Conseqüentemente, o seguinte conjunto de restrições pode ser escrito:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \quad \forall i = 1 \dots N \quad (4.2)$$

Considerando agora os conflitos, ao posicionar um rótulo em uma posição candidata, deve-se levar em consideração as sobreposições em potencial existentes, e com isso, um novo conjunto de restrições se faz necessário:

$$\begin{aligned} x_{i,j} + x_{k,t} - y_{i,j,k,t} &\leq 1 && \forall i = 1 \dots N \\ &&& \forall j = 1 \dots P_i \\ &&& (k,t) \in S_{i,j} \end{aligned} \quad (4.3)$$

Assim, a primeira formulação do PMNC pode ser escrita como:

$$v(PMNC^a) = Min \left\{ \sum_{i=1}^N \sum_{j=1}^{P_i} \left(w_{i,j} x_{i,j} + \sum_{(k,t) \in S_{i,j}} y_{i,j,k,t} \right) \right\} \quad (4.4)$$

(PMNC^a) Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \quad \forall i = 1 \dots N \quad (4.5)$$

$$\begin{aligned} x_{i,j} + x_{k,t} - y_{i,j,k,t} &\leq 1 & \forall i = 1 \dots N \\ & & \forall j = 1 \dots P_i \\ & & (k,t) \in S_{i,j} \end{aligned} \quad (4.6)$$

$$\begin{aligned} x_{i,j}, x_{k,t} \text{ e } y_{i,j,k,t} &\in \{0, 1\} & \forall i = 1 \dots N \\ & & \forall j = 1 \dots P_i \\ & & (k,t) \in S_{i,j} \end{aligned} \quad (4.7)$$

A restrição 4.7 garante que todas as variáveis de decisão são binárias. Ao minimizar a função objetivo 4.4, as variáveis de conflitos devem ser eliminadas ou minimizadas (caso não seja possível eliminá-las). A formulação 4.4-4.7 é parecida com a proposta por Zoraster (1990), porém permite rotular todos os pontos.

4.4 Modelagem Matemática Baseada em Pontos - PMNC^p

A segunda formulação matemática proposta para o PRCP, considera o grafo de conflitos formado por pontos com arestas indicando que alguma posição candidata do ponto i possui um potencial conflito com alguma posição candidata do ponto j .

Essa formulação foi inspirada no trabalho de Murray e Church (1996) que indica estratégias para reduzir formulações matemáticas. Dado que para cada ponto i somente uma posição candidata deverá ser usada (receberá o rótulo - restrição 4.5), a restrição de conflitos 4.6 pode ser alterada de modo a representar conflitos entre posições candidatas e pontos ao invés de outras posições candidatas.

Seja $C_{i,j}$ um conjunto com todos os pontos que possuem ao menos uma posição candidata em conflito com a posição candidata $x_{i,j}$, e $y_{i,j,c}$ um conflito entre a posição candidata $x_{i,j}$ e o ponto $c \in C_{i,j} : c > i$. Assim, as restrições 4.6 podem ser substituídas pelo seguinte conjunto de restrições:

$$|C_{i,j}|x_{i,j} + \sum_{(k,t) \in S_{i,j}} x_{k,t} - \sum_{c \in C_{i,j}} y_{i,j,c} \leq |C_{i,j}| \quad \forall i = 1 \dots N \quad (4.8)$$

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

Como as restrições 4.8 consideram variáveis de conflito que indicam sobreposições entre posições candidatas e pontos, a função objetivo 4.4 deve ser substituída por:

$$v(PMNC^p) = \text{Min} \left\{ \sum_{i=1}^N \sum_{j=1}^{P_i} \left(w_{i,j}x_{i,j} + \sum_{c \in C_{i,j}} y_{i,j,c} \right) \right\} \quad (4.9)$$

Assim, a segunda formulação do PMNC pode ser escrita como:

$$v(PMNC^p) = \text{Min} \left\{ \sum_{i=1}^N \sum_{j=1}^{P_i} \left(w_{i,j}x_{i,j} + \sum_{c \in C_{i,j}} y_{i,j,c} \right) \right\} \quad (4.10)$$

(PMNC^p) Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \quad \forall i = 1 \dots N \quad (4.11)$$

$$|C_{i,j}|x_{i,j} + \sum_{(k,t) \in S_{i,j}} x_{k,t} - \sum_{c \in C_{i,j}} y_{i,j,c} \leq |C_{i,j}| \quad \forall i = 1 \dots N \quad (4.12)$$

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

$$x_{i,j} \text{ e } y_{i,j,c} \in \{0, 1\} \quad \forall i = 1 \dots N \quad (4.13)$$

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

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

TABELA 4.1 – Número médio de restrições geradas.

# de Pontos	# de Instâncias	PMNC ^a	PMNC ^p
100	25	202	153
250	25	864	530
500	25	2909	1412
750	25	6181	2481
1000	25	10700	3643

A Tabela 4.1 apresenta o número médio de restrições por classe de problema, conforme os modelos $PMNC^a$ e $PMNC^p$. Os problemas aqui utilizados foram os propostos por Yamamoto et al. (2002) e estão disponíveis em <http://www.lac.inpe.br/~lorena/instancias.html>. A primeira coluna mostra o número de pontos, seguida do número de instâncias, número médio de restrições geradas pelo $PMNC^a$ e do número médio de restrições geradas pelo $PMNC^p$. Percebe-se que este último modelo reduz significativamente o número de restrições.

4.5 Relaxações Lagrangeanas Consideradas para o $PMNC^a$

O modelo definido em 4.4-4.7 quando aplicado a problemas com até 500 pontos e 4 posições candidatas forneceu, com auxílio do CPLEX 7.5 (ILOG, 2001a), as soluções ótimas em poucos segundos. Em contrapartida, para problemas maiores com 750 e 1000 pontos utilizados por Yamamoto e Lorena (2005), Alvim e Taillard (2005) e Cravo et al. (2007), os resultados ótimos não foram obtidos para algumas instâncias. Assim, além da LagClus, algumas relaxações lagrangeanas tradicionais foram propostas procurando resolver o $PMNC^a$.

4.5.1 Relaxação Lagrangeana Tradicional e LagSur

Semelhante ao que foi feito por Zoraster (1990), a primeira relaxação considerada foi uma relaxação lagrangeana sobre o conjunto de restrições 4.6. Assim, para um dado vetor de multiplicadores lagrangeanos $\lambda \in \mathfrak{R}_+^{NC}$, onde NC é o número restrições de conflito definido como:

$$NC = \sum_{i=1}^N \sum_{j=1}^{P_i} |S_{i,j}| \quad (4.14)$$

uma relaxação lagrangeana para o $PMNC^a$ pode ser definida como sendo:

$$v(L_\lambda PMNC^a) = Min \left\{ \sum_{i=1}^N \sum_{j=1}^{P_i} \left(w_{i,j} x_{i,j} + \sum_{(k,t) \in S_{i,j}} y_{i,j,k,t} \right) \right. \\ \left. \sum_{i=1}^N \sum_{j=1}^{P_i} \sum_{(k,t) \in S_{i,j}} \lambda_{i,j,k,t} (x_{i,j} + x_{k,t} - y_{i,j,k,t}) - \sum_{i=1}^N \sum_{j=1}^{P_i} \sum_{(k,t) \in S_{i,j}} \lambda_{i,j,k,t} \right\} \quad (4.15)$$

($L_\lambda PMNC^a$) Sujeito a 4.5 e 4.7.

O valor ótimo de $v(L_\lambda PMNC^a)$ é menor ou igual a $v(PMNC^a)$ e resulta da solução do

problema *dual* lagrangeano $Max_{\lambda \geq 0} \{v(L_{\lambda}PMNC^a)\}$, como foi visto no Capítulo 2.

Com relação ao algoritmo de subgradientes, em uma iteração k , os subgradientes são definidos como $g^{\lambda_k} = x_{i,j}^{\lambda_k} + x_{k,t}^{\lambda_k} - y_{i,j,k,t}^{\lambda_k} - 1$ com $i \in \{1, \dots, N\}$, $j \in \{1, \dots, P_i\}$ e $(k, t) \in S_{i,j}$, tal que $(x^{\lambda_k}, y^{\lambda_k})$ é a solução ótima de $L_{\lambda}PMNC^a$. O algoritmo de subgradientes implementado atualiza os multiplicadores lagrangeanos fazendo $\lambda_{k+1} = \lambda_k - \theta_k g^{\lambda_k}$ onde θ_k é o tamanho do passo calculado como $\theta_k = \frac{\pi(ub_k - lb_k)}{\|g^{\lambda_k}\|^2}$, sendo ub_k a melhor solução viável encontrada até a iteração k , e lb_k o melhor limitante dual encontrado também até a iteração k .

Uma solução relaxada composta de variáveis de posições candidatas e de variáveis de conflito, também é uma solução viável para $PMNC^a$ pois o conjunto de restrições 4.5 é respeitado. Essa solução viável é melhorada utilizando a heurística de busca local (heurística lagrangeana) mostrada na Figura 4.6.

Heurística de Melhoria - HM

1. Armazene em um vetor de conflitos o total de conflitos para cada posição candidata presente na solução viável;
2. **Para** $i=1$ **Até** o tamanho do vetor de conflitos **Faça**
3. **Se** Vetor de conflitos[i] $\neq 0$ **Então**
4. Procurar entre as possíveis posições candidatas j , a que apresenta o menor número de conflitos com a solução viável atual.
5. Se houver alguma posição candidata j com número de conflitos inferior a Vetor de conflitos[i], trocar a posição candidata presente em Solução viável[i] por j .
6. **Fim Para**

FIGURA 4.6 – Heurística lagrangeana de melhoria (HM) aplicada no algoritmo de subgradientes

A segunda relaxação considerada foi a Lagrangeana *Surrogate* (LagSur) (Narciso e Lorena, 1999).

Novamente o conjunto de restrições relaxadas foi o definido em 4.6. Primeiramente, é feita a relaxação no sentido *surrogate* como descrito por Glover (1968) sobre 4.6, e em seguida no sentido lagrangeano. Assim, para um dado vetor de multiplicadores $\lambda \in \mathfrak{R}_+^{NC}$, uma relaxação *surrogate* para $PMNC^a$ pode ser encontrada. Em seguida, ao se relaxar a restrição *surrogate* encontrada utilizando um multiplicador dual $t \geq 0$, pode-se obter a seguinte relaxação LagSur do $PMNC^a$:

$$v(LS_t PMNC^{a_1^\lambda}) = Min \left\{ \sum_{i=1}^N \sum_{j=1}^{P_i} \left(w_{i,j} x_{i,j} + \sum_{(k,t) \in S_{i,j}} y_{i,j,k,t} \right) + t \left(\sum_{i=1}^N \sum_{j=1}^{P_i} \sum_{(k,t) \in S_{i,j}} \lambda_{i,j,k,t} (x_{i,j} + x_{k,t} - y_{i,j,k,t}) - \sum_{i=1}^N \sum_{j=1}^{P_i} \sum_{(k,t) \in S_{i,j}} \lambda_{i,j,k,t} \right) \right\} \quad (4.16)$$

($LS_t PMNC^{a_1^\lambda}$) Sujeito a 4.5 e 4.7.

Uma característica interessante dessa relaxação é que para $t = 1$ obtém-se a relaxação lagrangeana mostrada anteriormente. Para o vetor λ fixo, o melhor valor para t pode ser calculado resolvendo-se o seguinte problema dual:

$$v(D_t^\lambda)_1 = Max_{t \geq 0} v(LS_t PMNC^{a_1^\lambda}) \quad (4.17)$$

O valor ótimo $v(D_t^\lambda)_1$, geralmente, é um limitante melhor que o obtido pela relaxação lagrangeana nas primeiras iterações (Senne e Lorena, 2000). Senne e Lorena (2000) descrevem um algoritmo de busca dicotômica que aproxima o melhor valor de t . Sendo t^* esse melhor valor, se para um dado número de iterações do algoritmo de subgradientes o valor de t^* repete, então esse valor é mantido fixo até o final do procedimento e a busca não é mais executada. Em uma iteração k , os multiplicadores são atualizados como descrito anteriormente para $L_\lambda PMNC^a$.

A LagSur também foi aplicada sobre o conjunto de restrições 4.5. Da mesma forma, dado um vetor de multiplicadores $\lambda \in \mathfrak{R}^N$, obtém-se uma relaxação *surrogate* para $PMNC^a$ que pode ser relaxada novamente no sentido lagrangeano utilizando um multiplicador dual t irrestrito, obtendo assim uma nova relaxação LagSur que pode ser expressa como:

$$v(LS_t PMNC^{a_2^\lambda}) = Min \left\{ \sum_{i=1}^N \sum_{j=1}^{P_i} \left(w_{i,j} x_{i,j} + \sum_{(k,t) \in S_{i,j}} y_{i,j,k,t} \right) + t \left(\sum_{i=1}^N \sum_{j=1}^{P_i} \lambda_i x_{i,j} - \sum_{i=1}^N \lambda_i \right) \right\} \quad (4.18)$$

($LS_t PMNC^{a_2^\lambda}$) Sujeito a 4.6 e 4.7.

Agora, os subgradientes são $g^{\lambda_k} = \sum_{j=1}^{P_i} x_{i,j}^{\lambda_k} - 1 \forall i = 1, \dots, N$, e o método de subgradientes atualiza os multiplicadores λ_k como mostrado anteriormente, porém a melhor solução viável agora considerada no cálculo do passo deve ser obtida pela heurística HM executada após uma heurística de construção como a mostrada na Figura 4.7. Isto porque,

ao relaxar as restrições 4.5 não existem garantias de que uma solução relaxada seja viável para o problema original.

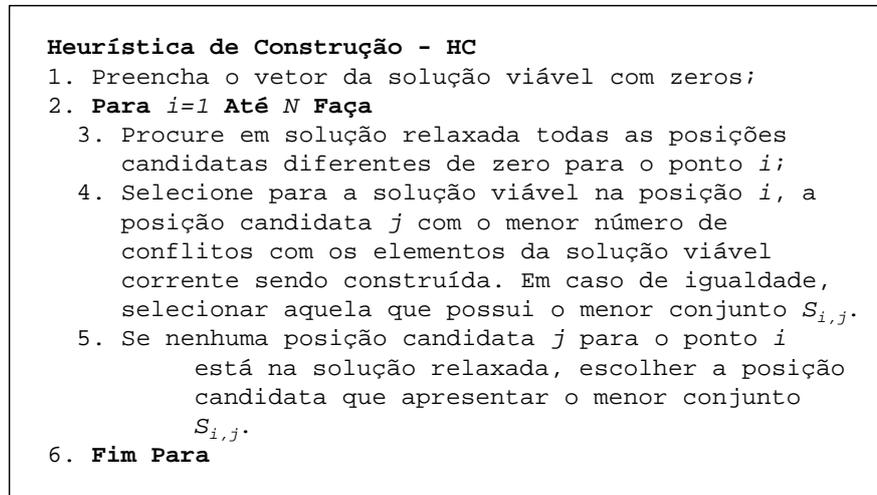


FIGURA 4.7 – Heurística de Construção (HC) aplicada no algoritmo de subgradientes.

4.5.2 LagClus para o PMNC^a

A LagClus para o PMNC^a foi aplicada conforme mostra o exemplo da Figura 4.8. Nesse exemplo, o grafo do problema é particionado em dois *clusters*. Ao se dividir são formados dois subproblemas menores que o problema original. Repare que neste grafo somente os conflitos entre posições candidatas são considerados como arestas.

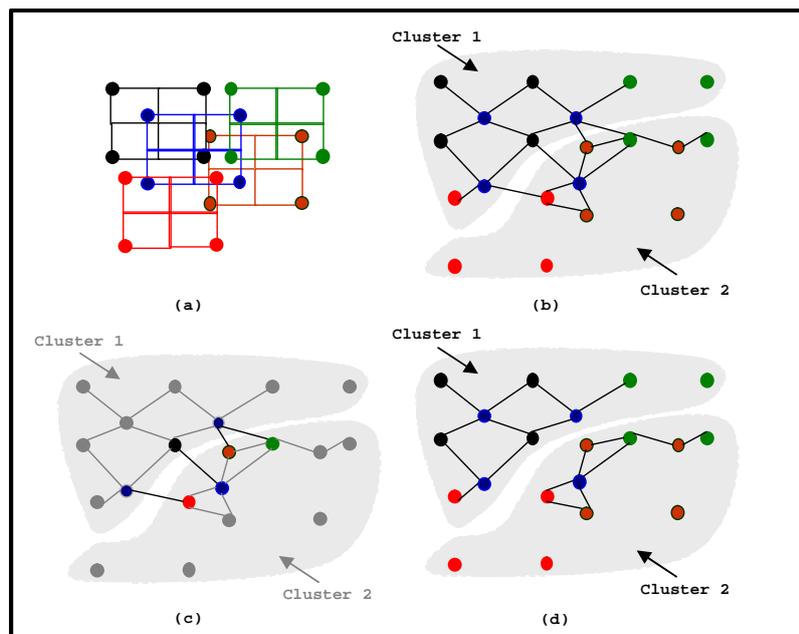


FIGURA 4.8 – Particionamento do grafo de conflitos. (a) Problema, (b) grafo de conflitos e os *clusters*, (c) arestas entre *clusters*, e (d) subproblemas gerados.

Sendo assim, a LagClus pode ser então aplicada da seguinte maneira no PRCP:

- a) Aplique uma heurística de particionamento para dividir G em \bar{P} partes, formando \bar{P} *clusters*;
- b) Com dois multiplicadores distintos, relaxe, no sentido lagrangeano, todas as restrições 4.5 e o grupo das restrições de conflito que correspondem às arestas entre *clusters*;
- c) A relaxação lagrangeana resultante pode ser decomposta em \bar{P} subproblemas e resolvida.

Reescrevendo a formulação de forma conveniente, o PMNC^a pode ser assim representado:

$$v(PMNC^a) = \text{Min} \left\{ \sum_{p=1}^{\bar{P}} (x^p + y^p) + \bar{y} \right\} \quad (4.19)$$

(PMNC^a) Sujeito a:

$$\begin{bmatrix} A_1 & A_2 & \cdots & A_{\bar{P}} & A_{\bar{y}} \\ D_1 & 0 & \cdots & 0 & 0 \\ 0 & D_2 & \cdots & 0 & 0 \\ & 0 & \ddots & & \\ 0 & 0 & \cdots & D_{\bar{P}} & 0 \end{bmatrix} \begin{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}^1 \\ \vdots \\ \begin{bmatrix} x \\ y \end{bmatrix}^{\bar{P}} \\ \bar{y} \end{bmatrix} \approx R \quad (4.20)$$

$$x^p \text{ e } y^p \in \{0, 1\}^{|V_p|} \quad \forall p \in \{1, \dots, \bar{P}\} \quad \text{e } \bar{y} \in \{0, 1\}^M \quad (4.21)$$

Sendo:

- x_p um vetor de variáveis de decisão (posições candidatas) associadas ao *cluster* p ;
- y_p um vetor de variáveis de conflito associadas às restrições de sobreposição presentes no *cluster* p ;
- \bar{y} um vetor de variáveis de conflito associadas às M restrições de conflitos que possuem vértices em diferentes *clusters*;

- A_p uma matriz que representa os coeficientes das variáveis associados ao *cluster* p que aparecem nas M restrições de conflito entre *clusters* e nas N restrições de igualdade 4.5;
- $A_{\bar{y}}$ uma matriz que representa os coeficientes das variáveis de conflito entre *clusters*;
- D^p uma matriz que representa os coeficientes das variáveis associadas ao *cluster* p que aparecem nas restrições de conflitos do *cluster* p ;
- R um vetor com os coeficientes do lado direito das restrições definidas em 4.5 e 4.6; e
- \approx operadores relacionais $=$ ou \leq que dependem das respectivas restrições.

Assim, relaxando no sentido lagrangeano as restrições geradas pelas matrizes A^p e $A_{\bar{y}}$, com os multiplicadores λ assim divididos: $\lambda_{1,2,\dots,N} \in \Re$ associados às N restrições de igualdade 4.5 e $\lambda_{N+1,N+2,\dots,N+M} \in \Re_+$ às M restrições de conflito 4.6 entre *clusters*, o modelo 4.19-4.21 pode ser decomposto em \bar{P} subproblemas tal que cada subproblema p pode ser assim descrito:

$$v(PMNC^a)_p = \text{Min} \{(1 + A^{Tp}\lambda)x^p + y^p : x^p \text{ e } y^p \in D_p\} \quad (4.22)$$

Portanto, a LagClus pode ser assim definida para o PMNC^a:

$$v(LG_\lambda PMNC^a) = \sum_{p=1}^{\bar{P}} v(PMNC^a)_p + \text{Min} \{(1 - A^{T\bar{y}}\lambda)\bar{y}\} - \sum_{m=1}^{N+M} R_m \lambda_m \quad (4.23)$$

Note que o termo $\text{Min} \{(1 - A^{T\bar{y}}\lambda)\bar{y}\}$ pode ser resolvido sem maiores problemas por inspeção. Os subgradientes agora devem ser calculados conforme o tipo de restrição e o algoritmo de subgradientes deve ser alterado para contemplar os dois tipos de multiplicadores aqui considerados: os associados às restrições de igualdade definidas em 4.5 que devem ser irrestritos e os associados às restrições de conflito 4.6 entre *clusters* que devem ser maiores ou iguais a zero.

4.6 Relaxações Lagrangeanas Consideradas para o PMNC^p

Como feito para o modelo baseado em posições candidatas, o CPLEX 7.5 com o modelo PMNC^p foi aplicado nas instâncias propostas por Yamamoto et al. (2002). Novamente, instâncias com até 500 pontos foram resolvidas em poucos segundos. Por outro lado, com instâncias maiores, o CPLEX 7.5 novamente não conseguiu encontrar a solução ótima dos problemas.

As relaxações da Seção 4.5, exceto a LagClus, foram aplicadas no modelo PMNC^p e os resultados foram semelhantes aos obtidos com o modelo PMNC^a.

Como o modelo PMNC^p é baseado em pontos, a LagClus pode ser explorada de uma outra maneira, conforme mostrado a seguir.

4.6.1 LagClus para o Modelo PMNC^p

A LagClus foi aplicada no PRCP utilizando o modelo PMNC^p, entretanto, como esta formulação é baseada em pontos, o particionamento foi feito de maneira diferente do apresentado na Seção 4.5.2. Considere a Figura 4.9 na qual em (a) é mostrado o problema e em (b) o grafo de conflitos obtido, assim como na Figura 4.8(a) e (b).

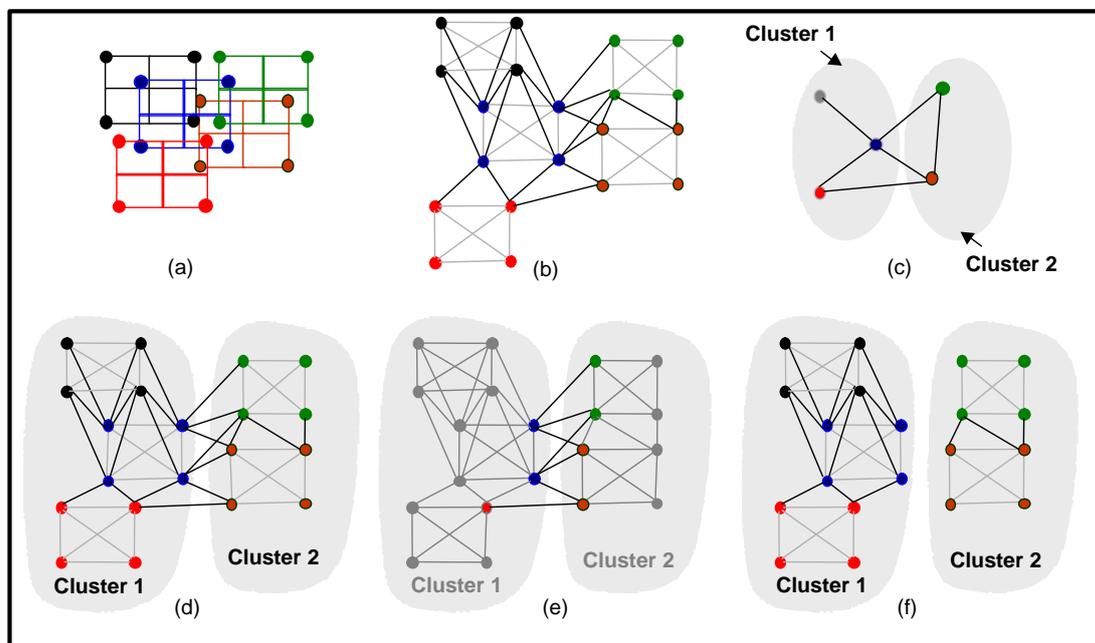


FIGURA 4.9 – Exemplo de aplicação da LagClus para a formulação baseada em pontos.

Porém, observando que as posições candidatas de cada ponto formam uma clique e que o PMNC^p é um modelo baseado em pontos, a técnica de contração de vértices (Goldbarg

e Luna, 2005) foi aplicada para gerar um grafo de conflitos entre pontos. Este método consiste em agrupar todas as posições candidatas do ponto i , formando um único vértice (ponto), veja Figura 4.9(c).

Aplicando sobre o grafo de conflitos entre pontos a técnica de particionamento de grafos, *clusters* são obtidos (veja Figura 4.9(c)). Após esta etapa, as contrações são expandidas gerando novamente o grafo de conflitos original (Figura 4.9(d)), surgindo às arestas entre *clusters* (Figura 4.9(e)) e os subproblemas (Figura 4.9(f)). Observe que esta técnica é semelhante ao processo de divisão utilizado nos algoritmos de particionamento de grafos multi níveis, apresentado no Capítulo 2.

A contração dos vértices permite que nos subproblemas apareçam as restrições 4.11, e com isso, não precisam ser relaxadas. Por outro lado, observa-se na Figura 4.9(e) as arestas entre “pontos” que devem ser relaxadas.

Como as restrições 4.11 são respeitadas, durante o algoritmo de subgradientes somente a heurística HM deve ser utilizada para melhorar uma solução relaxada dada pela LagClus.

4.6.2 Modelo Matemático da LagClus para o PMNC^p

Após o particionamento mostrado (contração - particionamento - expansão) e sendo \bar{P} o número de *clusters* obtidos, o PMNC^p pode ser escrito conforme o modelo 4.19-4.21 porém aqui A_p é uma matriz que representa os coeficientes das variáveis associados ao *cluster* p que aparecem nas M restrições de conflito entre *clusters*. Veja que não aparece as N restrições de igualdade definidas em 4.5, por que agora elas aparecem nas matrizes D_p . Note também que R agora é um vetor com os coeficientes do lado direito das restrições 4.11 e 4.12 ao invés de 4.5 e 4.6.

Novamente, as restrições geradas pelas matrizes A^p e $A_{\bar{y}}$ são relaxadas, mas com multiplicadores $\lambda \in \mathfrak{R}_+^M$. A relaxação obtida, denominada aqui LG _{λ} PMNC^p, pode ser decomposta em \bar{P} subproblemas, sendo cada um expresso matematicamente como mostrado em 4.22, consequentemente, a LagClus pode ser assim definida para o PMNC^p:

$$v(LG_{\lambda}PMNC^p) = \sum_{p=1}^{\bar{P}} v(PMNC^p)_p + Min \{ (1 - A^{T\bar{y}}\lambda)\bar{y} \} - \sum_{m=1}^M R_m \lambda_m \quad (4.24)$$

A Figura 4.10 ilustra como proceder com o modelo PMNC^p após a aplicação da LagClus. No exemplo mostrado, repare que, como a LagClus foi aplicada sobre o grafo de conflitos entre pontos, a formulação 4.10-4.13 pode ser aplicada nos *clusters* e com isso, somente

parte da restrição com vértices em diferentes *clusters* é relaxada. Compare a “Restrição com vértices em diferentes *clusters*” e a “Restrição a ser relaxada”, presentes na Figura 4.10.

4.7 Algoritmo de Geração de Colunas para o PMNC

Considerando a divisão do grafo de conflitos em \bar{P} *clusters* e aplicando a decomposição Dantzig-Wolfe (DW) sobre a relaxação de PL da formulação 4.19-4.21, obtém-se o seguinte PMR:

$$v(PMNC_{DW})_{PL} = Min \left\{ \sum_{p=1}^{\bar{P}} \sum_{j \in J_p} w^{j,p} \lambda_{j,p} + \bar{y} \right\} \quad (4.25)$$

(PMNC_{DW})_{PL} Sujeito a:

$$\sum_{p=1}^{\bar{P}} \sum_{j \in J_p} \lambda_{j,p} [A_p] \begin{bmatrix} x \\ y \end{bmatrix}^{j,p} + \bar{y} \approx R \quad (4.26)$$

$$\sum_{j \in J_p} \lambda_{j,p} = 1 \forall p \in \{1, \dots, \bar{P}\} \quad (4.27)$$

$$\lambda_{j,p} \geq 0 \forall p \in \{1, \dots, \bar{P}\}, j \in J_p \quad (4.28)$$

Sendo:

- $\lambda_{j,p}$ uma variável que representa o ponto extremo $j \in J_p$, ou seja, representa uma solução viável $\begin{bmatrix} x \\ y \end{bmatrix}^{j,p}$ do *cluster* p ;
- J_p um conjunto de pontos extremos do *cluster* p ; e
- $w^{j,p}$ o custo da solução x_j do subproblema p ;

Essa decomposição pode ser aplicada nos modelos PMNC^a e PMNC^p. Repare que o modelo 4.25-4.28 tem a mesma estrutura do modelo 3.20-3.23, entretanto aqui as restrições podem ser de igualdade e/ou menor ou igual, dependendo do modelo considerado (PMNC^a ou PMNC^p).

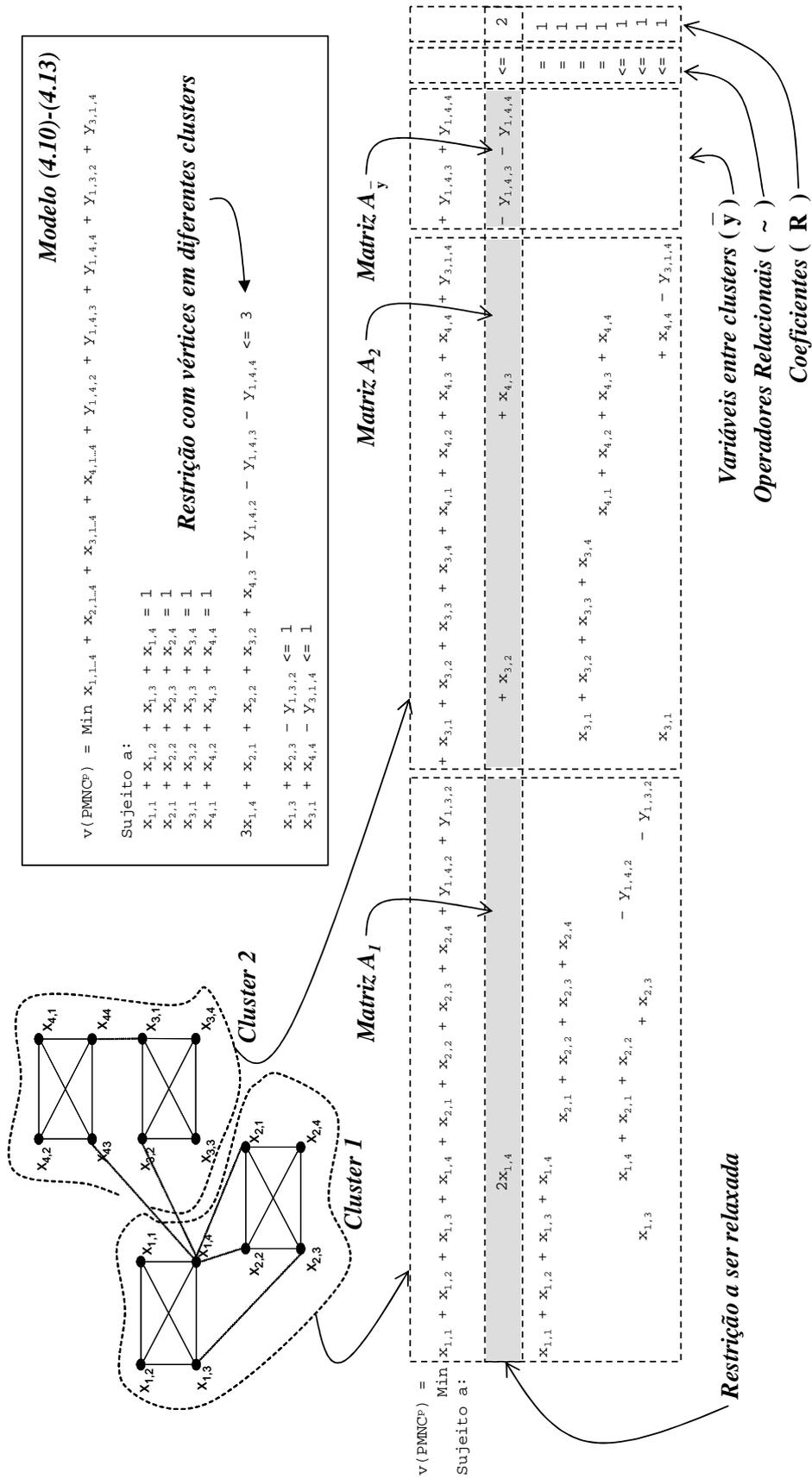


FIGURA 4.10 – Um exemplo para ilustrar as variáveis e matrizes presentes na LagClus para o modelo PMNCP.

Os \bar{P} subproblemas são os mesmos mostrados em 4.22, entretanto, os multiplicadores duais (Δ) correspondem às restrições 4.26:

$$v(PMNC)_p = \text{Min} \{ (1 + A_p^T \Delta) x^p + y^p : x^p \text{ e } y^p \in D_p \} \forall p = 1, \dots, \bar{P} \quad (4.29)$$

Para o PMR 4.25-4.28, uma nova coluna fornecida pelo p -ésimo *cluster* deve ser inserida no PMR se $v(PMNC)_p - \beta_p < 0$ sendo β_p a variável dual correspondente à p -ésima restrição de convexidade 4.27.

Sendo assim, a partir do PMR, as equações 4.23 e 4.24 podem ser assim reformuladas, respectivamente:

$$v(LG_{\Delta}PMNC^a) = \sum_{p=1}^{\bar{P}} v(PMNC^a)_p + \text{Min} \{ (1 - A^{T\bar{y}} \Delta) \bar{y} \} - \sum_{m=1}^{N+M} R_m \Delta_m \quad (4.30)$$

$$v(LG_{\Delta}PMNC^p) = \sum_{p=1}^{\bar{P}} v(PMNC^p)_p + \text{Min} \{ (1 - A^{T\bar{y}} \Delta) \bar{y} \} - \sum_{m=1}^M R_m \Delta_m \quad (4.31)$$

4.8 Resultados Computacionais

Os testes computacionais foram realizados com as instâncias propostas por Yamamoto et al. (2002), citadas anteriormente, em um computador equipado com um processador Pentium IV 2,66 GHz e 512 MB de memória RAM.

Como considerado em Zoraster (1990), Verner et al. (1997), Yamamoto e Lorena (2005), entre outros, em todos os problemas não foram consideradas preferências cartográficas, ou seja, considera-se custo ou penalidade igual a 1 para todas as posições candidatas, sendo o número total dessas posições igual a 4.

4.8.1 Resultados Computacionais Sem a Redução do Grafo de Conflitos

Inicialmente serão apresentados os resultados computacionais sem a redução proposta por Wagner et al. (2001). Essa estratégia permite avaliar o comportamento das relaxações e das decomposições de Dantzig-Wolfe, no pior caso, ou seja, com o grafo de conflitos completo.

4.8.1.1 Comparação entre os Modelos PMNC^a e PMNC^p

Inicialmente foram realizados experimentos com o CPLEX 7.5 considerando os dois modelos aqui apresentados, ou seja, PMNC^a e PMNC^p. Os resultados médios estão mostrados nas Tabelas 4.2 e 4.3, sendo que as colunas indicam:

- Prob. - Número de pontos que devem ser rotulados;
- LB - Melhor limite inferior (dual) encontrado;
- UB - Melhor limite superior encontrado (melhor solução viável encontrada);
- GAP (%) - Desvio percentual entre os limitantes superior e inferior calculado como: $Gap = \left(\frac{UB-LB}{UB} \right) \times 100$;
- # de Instâncias Resolvidas - Número de instâncias resolvidas de forma exata pelo CPLEX 7.5;
- Tempo (s) - Tempo computacional total médio em segundos;
- # de Rótulos Sobrepostos - Número médio de rótulos sobrepostos na melhor solução viável encontrada; e
- Proporção de Rótulos Livres (%) - Percentual médio de rótulos livres calculados como: $\left(\frac{N - \# \text{ de Rótulos Sobrepostos}}{N} \right) \times 100$.

TABELA 4.2 – Resultados médios do CPLEX 7.5 com o modelo PMNC^a.

Prob.	LB	UB	GAP (%)	# de Instâncias Resolvidas	Tempo (s)	# de Rótulos Sobrepostos	Proporção de Rótulos Livres (%)
100	100,00	100,00	0,00	25	0,02	0,00	100,00
250	250,00	250,00	0,00	25	0,06	0,00	100,00
500	500,84	500,84	0,00	25	3,12	1,68	99,67
750	756,13	759,36	0,42	9	6586,88	18,40	97,55
1000	1005,76	1048,88	4,11	0	5258,98	90,76	90,92

TABELA 4.3 – Resultados médios do CPLEX 7.5 com o modelo PMNC^p.

Prob.	LB	UB	GAP (%)	# de Instâncias Resolvidas	Tempo (s)	# de Rótulos Sobrepostos	Proporção de Rótulos Livres (%)
100	100,00	100,00	0,00	25	0,03	0,00	100,00
250	250,00	250,00	0,00	25	0,06	0,00	100,00
500	500,84	500,84	0,00	25	0,74	1,68	99,67
750	757,25	759,12	0,25	12	9625,92	17,84	97,62
1000	1010,37	1051,92	3,94	0	6683,80	97,12	90,29

O CPLEX 7.5 foi utilizado até resolver a instância ou parar por falta de memória.

Os resultados mostrados nas tabelas indicam que para instâncias de até 500 pontos, o CPLEX 7.5 fornece resultados semelhantes com os dois modelos. Porém, para instâncias com 750 e 1000 pontos, o modelo PMNC^p apresenta melhores resultados, resolvendo de forma ótima 12 instâncias das 25 propostas com 750 pontos, contra 9 do modelo PMNC^a (veja Tabela 4.2). Agora note que com 1000 pontos, o CPLEX 7.5 não encontrou nenhuma solução ótima.

A seguir, são apresentados os resultados das relaxações consideradas neste capítulo para o modelo PMNC^a.

4.8.1.2 Resultados das Relaxações Lagrangeana, LagSur e LagClus para o PMNC^a

O algoritmo de subgradientes descrito por Narciso e Lorena (1999) foi aqui implementado para resolver o dual lagrangeano de todas as relaxações. O controle utilizado para o parâmetro π (passo) foi o mesmo proposto por Held e Karp (1971). Inicia-se com o valor 2 e se durante 15 iterações o valor do limitante superior não decrescer, π é dividido pela metade. As condições de parada consideradas foram:

- a) $\pi \leq 0,005$;
- b) $(ub_k - lb_k) < 1$;
- c) $\|g^{\lambda_k}\|^2 = 0$; e
- d) o algoritmo de subgradientes também é finalizado quando uma solução sem conflitos é encontrada.

As Tabelas 4.4, 4.5 e 4.6 apresentam os resultados médios obtidos para L _{λ} PMNC^a e LS _{t} PMNC^a¹. As colunas indicam:

- v^* - Solução ótima do problema definido em 4.4-4.7, obtida usando o CPLEX 7.5;
- LB - Melhor limitante inferior (dual) encontrado;
- UB - Melhor limitante superior, ou seja, melhor solução viável encontrada;
- Gap UB (%) - Desvio percentual do melhor limitante superior encontrado em relação à solução ótima, calculado como: $Gap \text{ UB} = \left(\frac{UB-v^*}{v^*}\right) * 100$;

- Gap LB (%) - Desvio percentual do melhor limitante inferior encontrado em relação à solução ótima, calculado como: $Gap\ LB = \left(\frac{v^* - LB}{v^*}\right) * 100$;
- Iterações - Número de iterações utilizadas pelo algoritmo de subgradientes;
- Tempo (s) - Tempo computacional total em segundos.

A procura pelo melhor multiplicador t descrita por Senne e Lorena (2000) conduziu ao valor zero em $LS_t PMNC^{a\hat{\lambda}}$. Esse fato era esperado, pois toda informação sobre os conflitos (variáveis $y_{i,j,jk}$) foi relaxada. Com isso, alguns valores para t foram pré-fixados: 0,25; 0,5 e 0,75.

Para os problemas com 750 e 1000 pontos, as soluções ótimas não foram obtidas. Instâncias com 100 e 250 pontos são simples, uma solução totalmente sem conflito é encontrada rapidamente, com isso, o algoritmo de subgradientes pára em limitantes inferiores ruins, conseqüentemente os limitantes inferiores e os *Gaps* não foram calculados, sendo substituídos por NC (Não calculado). A mesma situação ocorreu em algumas instâncias com 500 pontos. Portanto, os resultados mostrados nas Tabelas 4.4, 4.5 e 4.6 correspondem aos problemas em que o processo de otimização encontra uma das três primeiras condições de parada do algoritmo de subgradientes.

TABELA 4.4 – Resultados médios para o limitante inferior das relaxações lagrangeana e LagSur sobre as restrições 4.6.

Prob.	v^*	LB - Limitante Inferior				GAP LB (%)			
		$L_\lambda PMNC^a$	$LS_t PMNC^{a\hat{\lambda}}$			$L_\lambda PMNC^a$	$LS_t PMNC^{a\hat{\lambda}}$		
			$t = 0,25$	$t = 0,50$	$t = 0,75$		$t = 0,25$	$t = 0,50$	$t = 0,75$
100	100,00	NC	NC	NC	NC	NC	NC	NC	NC
250	250,00	NC	NC	NC	NC	NC	NC	NC	NC
500	500,84	498,20	493,46	496,91	499,45	0,53	1,48	0,80	0,28
750	-	749,98	743,42	748,74	749,98	-	-	-	-
1000	-	999,93	997,04	999,93	999,95	-	-	-	-

TABELA 4.5 – Resultados médios para o limitante superior das relaxações lagrangeana e LagSur sobre as restrições 4.6.

Prob.	v^*	UB - Limitante Superior				GAP UB (%)			
		$L_\lambda PMNC^a$	$LS_t PMNC^{a\hat{\lambda}}$			$L_\lambda PMNC^a$	$LS_t PMNC^{a\hat{\lambda}}$		
			$t = 0,25$	$t = 0,50$	$t = 0,75$		$t = 0,25$	$t = 0,50$	$t = 0,75$
100	100,00	100,00	100,00	100,00	100,00	0,00	0,00	0,00	0,00
250	250,00	250,00	250,00	250,04	250,00	0,00	0,00	0,02	0,00
500	500,84	503,92	502,88	503,16	503,80	0,61	0,41	0,46	0,59
750	-	774,44	771,80	772,68	774,44	-	-	-	-
1000	-	1086,44	1075,64	1079,24	1086,40	-	-	-	-

Considerando a Tabela 4.4, os *Gaps* dos limitantes inferiores variaram de 0,28% a 1,48%. Na Tabela 4.5, os *Gaps* encontrados para os limitantes superiores variaram de 0,00% a 0,61%. $LS_tPMNC^{a\lambda}$ apresentou resultados melhores para os dois *Gaps* avaliados. Os tempos computacionais variaram de 0 a 19,60s para ambas as relaxações (veja Tabela 4.6).

Os resultados obtidos para $LS_tPMNC^{a\lambda}$ estão mostrados na Tabela 4.7. O CPLEX 7.5 foi utilizado para resolver os problemas de programação linear inteira binária.

O limitante da relaxação $LS_tPMNC^{a\lambda}$ é mais “forte” que os fornecidos pelas relaxações $L_\lambda PMNC^a$ e $LS_tPMNC^{a\lambda}$, pois toda informação sobre conflitos permanece no problema relaxado. No entanto, os resultantes não foram interessantes. Somente o limitante inferior foi melhorado, e os tempos computacionais aumentaram consideravelmente. Os limitantes superiores não foram melhores, e os problemas com 500, 750 e 1000 pontos não puderam ser resolvidos em 7200 segundos de execução do algoritmo de subgradientes.

TABELA 4.6 – Iterações e tempos computacionais das relaxações lagrangeana e LagSur sobre as restrições 4.6.

Prob.	Iterações				Tempo (s)				
	$L_\lambda PMNC^a$	$LS_tPMNC^{a\lambda}$			v^*	$L_\lambda PMNC^a$	$LS_tPMNC^{a\lambda}$		
		$t = 0, 25$	$t = 0, 50$	$t = 0, 75$			$t = 0, 25$	$t = 0, 50$	$t = 0, 75$
100	1,16	1,16	1,16	1,16	0,02	0,00	0,00	0,00	0,04
250	2,48	2,36	7,72	2,68	0,06	0,00	0,00	0,04	0,00
500	146,00	140,24	134,48	146,20	3,12	2,76	2,64	2,64	2,88
750	146,00	146,00	146,00	146,00	-	8,48	8,44	8,40	8,56
1000	146,00	146,00	146,00	146,00	-	19,32	19,60	19,52	19,08

TABELA 4.7 – Iterações e tempos computacionais da LagSur sobre as restrições 4.5.

Prob.	v^*	LB	UB	Gap UB (%)	Gap LB (%)	Iter.	Tempo (s)	
							v^*	$LS_tPMNC^{a\lambda}$
100	100	NC	100	0,00	NC	1,00	0,02	0,16
250	250	NC	250	0,00	NC	1,48	0,06	0,92

A última relaxação aplicada para o $PMNC^a$ foi a LagClus. A Tabela 4.8 mostra informações referentes ao particionamento para cada classe de problemas. A primeira coluna diz respeito ao problema, seguida do número de vértices presentes no grafo de conflitos, do número de *clusters* considerado, e por último de forma aproximada, do número de vértices presentes em cada *cluster*.

Na Tabela 4.9 estão mostrados os resultados obtidos com a LagClus. O particionamento foi realizado com o METIS de Karypis e Kumar (1998b), e os subproblemas foram resolvidos usando o CPLEX 7.5. A coluna “Gap %” foi calculada na seguinte maneira: $Gap = \left(\frac{UB-LB}{UB}\right) \times 100$.

TABELA 4.8 – Informações sobre o particionamento realizado para a LagClus com o modelo PMNC^a.

Prob.	# de Posições Candidatas	# de Clusters	# Médio de Vértices em cada cluster
100	400	4	100
250	1000	10	100
500	2000	20	100
750	3000	25	120
1000	4000	60	≈67

TABELA 4.9 – Resultados médios obtidos com a LagClus sobre o PMNC^a.

Prob.	LB	UB	GAP LB %	GAP UB %	GAP (%)	# de Instâncias Resolvidas	Tempo (s)	# de Rótulos Sobrepostos	Proporção de Rótulos Livres (%)
100	100,00	100,00	0,00	0,00	0,00	25	0,16	0,00	100,00
250	250,00	250,00	0,00	0,00	0,00	25	2,36	0,00	100,00
500	498,43	501,52	0,76	1,14	0,61	25	82,72	3,08	99,38
750	749,41	767,08	-	-	2,30	0	337,80	33,56	95,53
1000	1002,11	1070,60	-	-	6,39	0	817,00	135,32	86,47

Os problemas com 750 e 1000 pontos, foram resolvidos em média em 337,80 e 817,00s, respectivamente. Pode-se observar que os limitantes inferiores e superiores foram, em geral, melhores que os obtidos pelas demais relaxações. Os *Gaps* superiores variaram de 0,00% a 1,14%, e os inferiores de 0,00% a 0,76%. Para instâncias com 1000 pontos, o limitante trivial imposto pelo número de pontos, foi ultrapassado.

Se a Tabela 4.9 for comparada com as Tabelas 4.2 e 4.3, o CPLEX 7.5 apresenta limitantes melhores que a LagClus para os problemas com 750 e 1000 pontos, porém em um tempo computacional maior. Como o CPLEX 7.5 com o modelo PMNC^p apresentou os melhores resultados, é de se esperar que a LagClus para este modelo forneça soluções melhores que a LagClus com o modelo PMNC^a.

4.8.1.3 Resultados da LagClus para o PMNC^p

Como mencionado na Seção 4.6, as relaxações $L_\lambda \tilde{\text{PMNC}}^p$, $LS_t \text{PMNC}^{p_1^\lambda}$ e $LS_t \text{PMNC}^{p_2^\lambda}$ apresentaram comportamentos semelhantes dos mostrados nas Tabelas 4.4, 4.5, 4.6 e 4.7, consequentemente optou-se por não apresentá-los aqui.

A Tabela 4.10 apresenta informações sobre o particionamento da LagClus para o PMNC^p. Note que o total de *clusters* adotado para cada tipo de problema, está diferente do apontado na Tabela 4.8, isso porque no PMNC^p o grafo particionado é um grafo de conflitos entre pontos que é bem menor que o de posições candidatas do modelo PMNCP^a.

TABELA 4.10 – Informações sobre o particionamento realizado para a LagClus com o modelo PMNC^p.

Prob.	# de Posições Candidatas	# de Clusters	# Médio de Vértices em cada cluster
100	400	2	200
250	1000	2	500
500	2000	2	1000
750	3000	10	300
1000	4000	25	160

TABELA 4.11 – Resultados médios obtidos com a LagClus sobre o PMNC^a.

Prob.	LB	UB	GAP LB %	GAP UB %	Gap (%)	# de Instâncias Resolvidas	Tempo (s)	# de Rótulos Sobrepostos	Proporção de Rótulos Livres (%)
100	100,00	100,00	0,00	0,00	0,00	25	0,12	0,00	100,00
250	250,00	250,00	0,00	0,00	0,00	25	0,12	0,00	100,00
500	500,84	500,84	0,00	0,00	0,00	25	0,40	1,68	99,67
750	758,09	758,96	-	-	0,12	21	53,84	17,60	97,65
1000	1030,07	1047,32	-	-	1,64	0	3445,40	90,16	90,98

TABELA 4.12 – Resultados médios da LagClus sobre o PMNC^p variando o número de *clusters* para os problemas com 1000 pontos.

Número de clusters	LB	UB	GAP (%)	Tempo (s)	# de Rótulos Sobrepostos	Proporção de Rótulos Livres
20	1031,23	1044,80	1,30	3842,84	85,80	91,42
25	1030,07	1047,32	1,64	3445,40	90,16	90,98
30	1026,81	1049,16	2,13	734,80	93,56	90,64

A Tabela 4.11 apresenta os resultados da LagClus para o PMNC^p. Percebe-se que esses resultados são melhores que todos os apresentados anteriormente, inclusive melhores que os do CPLEX 7.5.

Tentando resolver de forma ótima as instâncias com 1000 pontos e mostrar o comportamento da LagClus quando o número de *clusters* varia, novos experimentos foram realizados e os resultados estão mostrados na Tabela 4.12. Como esperado, quando o número de *clusters* é reduzido, o tempo do algoritmo de subgradientes aumenta (os

subproblemas se tornam mais complicados de serem resolvidos), porém os limitantes fornecidos são melhores. Agora de maneira contrária, quando o número de *clusters* aumenta, o tempo do algoritmo de subgradientes reduz e os limitantes se tornam ruins.

4.8.2 Resultados Computacionais Com a Redução do Grafo de Conflitos

A Tabela 4.13 mostra a redução imposta aos problemas após o método de redução de Wagner et al. (2001). As colunas indicam valores médios do número de vértices ($|V|$), de arestas ($|A|$), de pontos fixados pelo processo (Pontos rotulados), do percentual relativo à redução de vértices (Redução de vértices (%)) e de arestas (Redução de arestas (%)).

O método de Wagner et al. (2001) foi mais efetivo nas instâncias com 100 e 250 pontos, gerando em muitos casos, soluções sem conflitos.

TABELA 4.13 – Informações sobre o processo de redução do grafo de conflitos.

Prob.	Grafo de Conflitos Original		Grafo de Conflitos Reduzido		
	$ V $	$ A $	Pontos Rotulados	Redução de Vértices	Redução de Arestas
100	400	102,28	98,52	98,52	90,66
250	1000	614,00	229,60	91,84	74,95
500	2000	2409,44	344,36	68,87	46,43
750	3000	5431,68	347,84	46,38	29,08
1000	4000	9700,92	276,36	27,64	16,24

4.8.2.1 Resultados Computacionais Com a Redução do Grafo de Conflitos

Como os resultados computacionais mostraram que o modelo PMNC^p fornece resultados melhores que o PMNC^a, os testes aqui apresentados consideram somente o modelo PMNC^p.

TABELA 4.14 – Resultados médios do CPLEX 10 com o modelo PMNC^p usando grafos reduzidos.

Prob.	LB	UB	GAP (%)	# de Instâncias Resolvidas	Tempo (s)	# de Rótulos Sobrepostos	Proporção de Rótulos Livres (%)
100	100,00	100,00	0,00	25	0,00	0,00	100,00
250	250,00	250,00	0,00	25	0,01	0,00	100,00
500	500,84	500,84	0,00	25	0,14	1,68	99,67
750	757,10	758,92	0,24	14	5517,12	17,60	97,65
1000	1004,87	1042,88	3,64	0	4504,52	83,12	91,69

Sendo assim, a Tabela 4.14 mostra os resultados do CPLEX 10 (ILOG, 2006b) com os

grafos reduzidos. Novamente, o CPLEX foi utilizado até resolver a instância ou parar por falta de memória.

Considerando a Tabela 4.14 note que, mesmo com uma versão mais recente do *solver* e grafos reduzidos, o CPLEX ainda não resolve nenhuma instância com 1000 pontos e apenas 14 com 750 pontos. A proporção de rótulos livres aumentou em relação aos demais experimentos apresentados, mas por outro lado, o tempo computacional continua elevado, variando de 0,00 a 5517,12 segundos.

4.8.2.2 Resultados da LagClus para o PMNC^p Usando Grafos Reduzidos

Como a LagClus apresentou melhores resultados que as demais relaxações, optou-se somente em utilizá-la sobre o grafo de conflitos reduzido.

Considerando o mesmo número de *clusters* apresentados na Tabela 4.10, a Tabela 4.15 apresenta os resultados da LagClus para o PMNC^p utilizando grafos reduzidos, sendo que os subproblemas foram resolvidos com o CPLEX 10.

TABELA 4.15 – Resultados médios da LagClus com o modelo PMNC^p sobre os grafos reduzidos.

Prob.	LB	UB	GAP LB %	GAP UB %	Gap (%)	# de Instâncias Resolvidas	Tempo (s)	# de Rótulos Sobrepostos	Proporção de Rótulos Livres (%)
100	100,00	100,00	0,00	0,00	0,00	25	0,02	0,00	100,00
250	250,00	250,00	0,00	0,00	0,00	25	0,04	0,00	100,00
500	500,84	500,84	0,00	0,00	0,00	25	0,12	1,68	99,67
750	758,92	758,92	0,00	0,00	0,00	25	4,57	17,56	97,66
1000	1036,93	1047,32	-	-	0,21	7	1238,82	76,10	92,39

Assim como aconteceu com o CPLEX, a LagClus também forneceu melhores soluções com grafos reduzidos. Para problemas com 750 pontos, ela resolveu todas as instâncias (25) contra 21 da LagClus sobre o grafo original (Tabela 4.11) e 14 com o CPLEX 10 sobre o grafo reduzido (Tabela 4.14).

Considerando agora os resultados encontrados para os problemas com 1000 pontos, a LagClus com grafos reduzidos foi capaz de encontrar a solução ótima de 7 instâncias contra nenhuma do CPLEX (Tabela 4.14) e nenhuma da LagClus sobre o grafo original (Tabela 4.11). Repare também que o tempo computacional foi menor e que a proporção de rótulos livres aumentou.

4.8.3 Resultados Computacionais do AGC

Nesta seção são apresentados os resultados computacionais do AGC com o modelo $PMNC^p$ e grafos reduzidos, uma vez que dessa maneira, a LagClus produziu os melhores limitantes.

A Figura 4.11 descreve o processo de geração de colunas utilizado que se encerra quando nenhuma outra coluna é inserida no PMR. Note que após o processo de geração de colunas, o PMR é transformado em um problema inteiro e resolvido com o CPLEX. Conforme Desrosiers e Lübbecke (2005), este processo pode ser considerado como sendo uma heurística.

Para o AGC, foi utilizado o mesmo particionamento mostrado na Tabela 4.10 e tanto os subproblemas quanto o PMR, foram resolvidos usando o CPLEX 10.

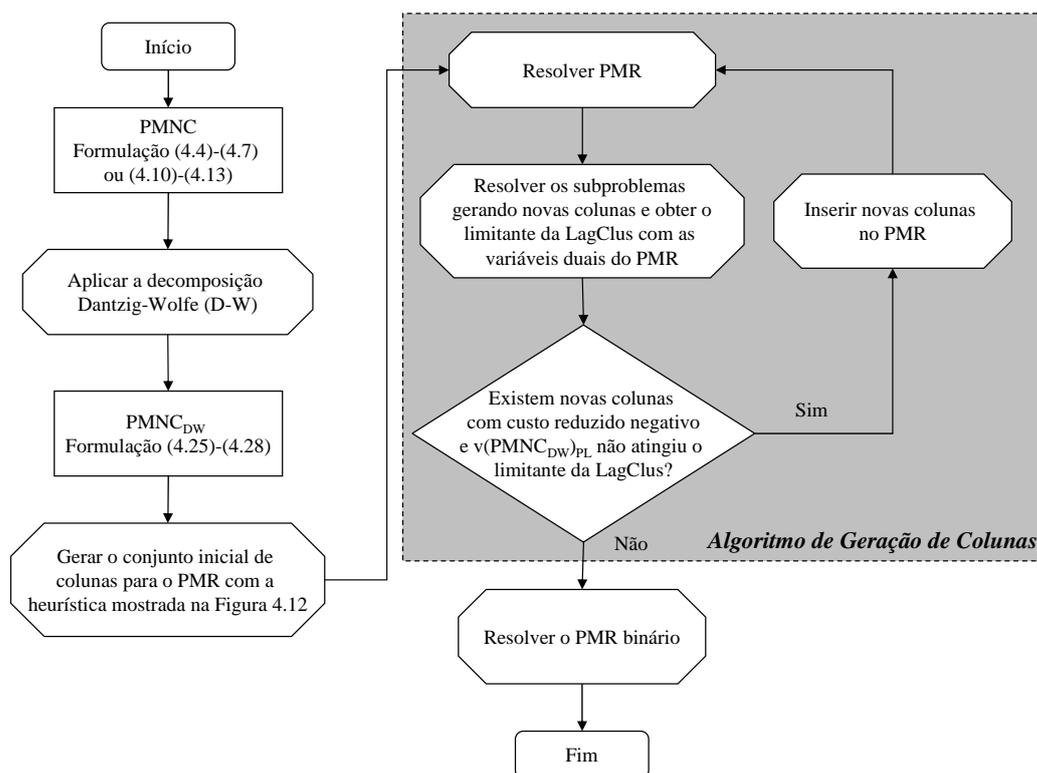


FIGURA 4.11 – Diagrama dos passos da AGC para o $PMNC^p$.

O conjunto inicial de colunas foi composto da seguinte maneira. Após gerar uma solução viável aleatória para o PMNC, a heurística HM é aplicada e a solução final gerada, é decomposta nos *clusters* e introduzida no PMR. O processo então é repetidamente utilizado até compor o conjunto inicial de colunas desejado. Esse algoritmo está mostrado na Figura 4.12.

Note que o valor da melhor solução encontrada é armazenado no algoritmo mostrado na Figura 4.12. Isso permite verificar se realmente o processo de geração de colunas está encontrando colunas melhores, uma vez que no final do processo, o PMR binário é resolvido e uma solução viável para o PMNC é obtida.

```

Algoritmo Inserir_colunas_no_RMP(Soluções_Iniciais)
// Seja:
// - Solução_Gerada_Aleatoriamente(N) uma função que gera uma
// solução aleatória para N pontos
// - Função_Objetivo (Sol) a função objetivo do PMNC
// - Inserir_no_PMR(Sol) uma função que insere Sol no PMR
1 Melhor_Solução ← ∞
2 Para i_sol=1 Até Soluções_Iniciais Faça
3 Sol ← Solução_Gerada_Aleatoriamente (N)
4 Bkp_Sol ← Sol
5 Encontrou_Nova_Solução ← verdadeiro HM
6 fAtual ← Função_Objetivo(Bkp_Sol)
7 Bkp_Atual ← fAtual
8 Enquanto Encontrou_Nova_Solução Faça
9 Encontrou_Nova_Solução ← falso
10 Para i = 1 Até N Faça
11 Posição_Candidata_Atual = Bkp_Sol [i]
12 Para  $\forall j \in P_i$  Faça
13 Se j == Bkp_Sol [i] Então Continue
14 Bkp_Sol [i] ← j
15 Se (Função_Objetivo(Bkp_Sol) < fAtual) Então
16 fAtual ← Função_Objetivo (Bkp_Sol)
17 Encontrou_Nova_Solução ← verdadeiro
18 Melhor_Vizinho ← j
19 Ponto_Alterado ← i
20 Fim Se
21 Fim Para
22 Bkp_Sol [i] ← Posição_Candidata_Atual
23 Fim Para
24 Se Encontrou_Nova_Solução Então
25 Bkp_Sol [Ponto_Alterado] ← Melhor_Vizinho
26 Fim Se
27 Fim Enquanto
28 Se (fAtual < Bkp_Atual) E (fAtual != Melhor_Solução) Então
29 Sol ← Bkp_Sol
30 Se (fAtual < Melhor_Solução) Então
31 Melhor_Solução ← fAtual
32 Fim Se
33 Fim Se
34 Inserir_no_PMR(Sol)
35 Fim Para

```

FIGURA 4.12 – Algoritmo para gerar o conjunto inicial de colunas.

A Tabela 4.16 apresenta os resultados do AGC. Nesta tabela, algumas colunas não foram ainda apresentadas neste capítulo, são elas:

- Melhor Solução Inicial - Melhor solução inicial adicionada no PMR pelo algoritmo descrito na Figura 4.12;
- # Inicial de Colunas - Número inicial de colunas no PMR após gerar 1000 soluções viáveis com o algoritmo descrito na Figura 4.12;
- PMR Inicial - Valor inicial de $v(PMNC_{DW})_{PL}$ com o conjunto inicial de colunas;
- # Final de Colunas - Número final de colunas no PMR;

- PMR Final - Valor final do $v(PMNC_{DW})_{PL}$ com todas as colunas;
- Tempo₁ (s) - Tempo em segundos utilizado pelo processo de geração de colunas;
- PMR Binário - Valor do $v(PMNC_{DW})$ com todas as variáveis binárias; e
- Tempo₂ (s) - Tempo em segundos utilizado pelo CPLEX 10 para resolver o PMR binário.

Os resultados mostrados na Tabela 4.16 são interessantes. O algoritmo gera um pequeno conjunto de colunas. No pior caso, foram aproximadamente 52 novas colunas para os problemas com 1000 pontos. O tempo computacional variou de 0,00 a 84,04 segundos para o AGC e de 0,00 a 2,64 segundos para o PMR binário, sendo assim, no pior caso, o AGC levou 86,68 segundos para concluir todo o processo descrito na Figura 4.11.

Observe agora a qualidade da solução encontrada. Tomando como exemplo os problemas com 1000 pontos, note que a melhor solução inicial adicionada no PMR tem uma função objetivo igual a 1123,24 (lembre-se que se trata de uma média entre 25 instâncias). Por outro lado, veja que a melhor solução encontrada foi de 1039,04. Isso mostra que o PMR realmente está guiando seus subproblemas de tal maneira que novas colunas “boas” sejam encontradas.

TABELA 4.16 – Resultados médios da LagClus com o modelo PMNC^p sobre os grafos reduzidos.

Prob.	AGC						PMR Binário		# de Instâncias Resolvidas	# de Rótulos Sobrepostos	Proporção de Rótulos Livres (%)
	Melhor Solução Inicial	# Inicial de Colunas	PMR Inicial	# Final de Colunas	PMR Final	Tempo ₁ (s)	PMR Binário	Tempo ₂ (s)			
100	100,04	2000	100,00	2002	100,00	0,00	100,00	0,08	25	0,00	100,00
250	250,84	2000	250,00	2002	250,00	0,00	250,00	0,12	25	0,00	100,00
500	504,00	2000	502,84	2003,04	500,84	0,16	500,84	0,00	25	1,64	99,67
750	785,52	10000	765,16	10013,40	758,92	14,48	758,92	0,48	25	17,50	97,67
1000	1123,24	25000	1048,97	25052,44	1037,56	84,04	1039,04	2,64	10	76,04	92,40

Como os coeficientes da função objetivo 4.25 são inteiros, uma solução dada pelo PMR binário pode ser considerada ótima se a diferença entre os valores das funções objetivos do PMR binário e PMR final for menor que 1, uma vez que o limitante inferior do AGC é equivalente ao da LagClus, veja equação 4.31.

Sendo assim, o AGC encontrou todas as soluções ótimas para as instâncias com 100, 250, 500 e 750 pontos. Para problemas com 1000 pontos, o AGC encontrou 10 soluções ótimas contra 7 da LagClus e nenhuma do CPLEX 10. Ao analisar os resultados sobre

rótulos com sobreposição e proporção de rótulos livres, novamente o AGC foi melhor que os demais métodos.

4.8.4 Comparação com a Literatura

Os melhores resultados aqui encontrados foram comparados com as melhores soluções conhecidas na literatura descrita nos trabalhos de Christensen et al. (1995), Verner et al. (1997) e Yamamoto e Lorena (2005). A Tabela 4.17 apresenta a proporção de rótulos livres encontrados aqui e os demais apresentados na literatura.

Mais uma vez lembre-se que existem três abordagens para o PRCP com objetivos diferentes. Como pode ser observado na Tabela 4.17, o AGC forneceu bons resultados. Os tempos computacionais desses algoritmos não foram comparados, pois os testes foram realizados em máquinas diferentes.

TABELA 4.17 – Comparação com outros algoritmos.

Algoritmo	Proporção de Rótulos Livres (%)				
	Problemas				
	100	250	500	750	1000
AGC	100,00	100,00	99,67	97,67	92,40
LagClus	100,00	100,00	99,67	97,66	92,39
GRASP Melhor (Cravo et al., 2007)	100,00	100,00	99,67	97,72	92,20
POPMUSIC (Alvim e Taillard, 2005)	100,00	100,00	99,66	97,64	92,18
AGC Melhor (Yamamoto e Lorena, 2005)	100,00	100,00	99,60	97,10	90,70
AGC Média (Yamamoto e Lorena, 2005)	100,00	100,00	99,60	96,80	90,40
Busca Tabu (Yamamoto et al., 2002)	100,00	100,00	99,30	96,80	90,00
AG com máscara (Verner et al., 1997)	100,00	99,98	98,79	95,99	88,96
AG sem máscara (Verner et al., 1997)	100,00	98,40	92,59	82,38	65,70
<i>Simulated Annealing</i> (Christensen et al., 1995)	100,00	99,90	98,30	92,30	82,09
Zoraster (1990)	100,00	99,79	96,21	79,78	53,06
Hirsh (1982)	100,00	99,58	95,70	82,04	60,24
<i>3-opt Gradient Descent</i> (Christensen et al., 1995)	100,00	99,76	97,34	89,44	77,83
<i>2-opt Gradient Descent</i> (Christensen et al., 1995)	100,00	99,36	95,62	85,60	73,37
<i>Gradient Descent</i> (Christensen et al., 1995)	98,64	95,47	86,46	72,40	58,29
<i>Greedy Algorithm</i> (Christensen et al., 1995)	95,12	88,82	75,15	58,57	43,41

4.9 Considerações Finais

Este capítulo apresentou os resultados da LagClus e do AGC para o Problema da Rotulação Cartográfica de Pontos (PRCP) considerando a abordagem de minimizar o número de conflitos. Primeiramente foram apresentados dois novos modelos para o PRCP, um baseado em arestas (PMNC^a) e outro baseado em pontos (PMNC^p). Este último apresenta um número menor de restrições mais fortes do que as do modelo PMNC^a.

Várias relaxações foram consideradas para os dois modelos, dentre elas lagrangeana tradicional e LagSur. Entretanto, estas relaxações não conseguiram fornecer bons limitantes, muitas delas não ultrapassaram o limitante trivial imposto pelo número de

pontos a serem rotulados.

A LagClus foi então testada nas duas formulações. Resultados parciais com o grafo de conflitos original mostraram que essa relaxação realmente produz um limitante dual de melhor qualidade. Os *gaps* obtidos foram pequenos e várias instâncias foram resolvidas de forma ótima.

Os resultados da LagClus foram ainda melhores quando foi aplicada a técnica de redução proposta por Wagner et al. (2001). Neste caso, instâncias consideradas difíceis com 1000 pontos puderam ser resolvidas de forma ótima.

O AGC pareceu ser a melhor estratégia para o PRCP, pois forneceu resultados melhores que a LagClus em um tempo computacional menor. A proporção de rótulos livres fornecido pelos dois métodos (LagClus e AGC), foi melhor que alguns algoritmos presentes na literatura.

O próximo capítulo apresenta os resultados da LagClus e do AGC em um problema bem conhecido na literatura de corte e empacotamento.

CAPÍTULO 5

CARREGAMENTO DE PALETES DO PRODUTOR E ESTIVAGEM DE UNIDADES DE CELULOSE

Este capítulo tem como objetivo apresentar os resultados da LagClus aplicada ao Problema do Carregamento de Paletes do Produtor (PCPP) e ao Problema da Estivagem de Unidades de Celulose (PEUC). O PEUC é um problema tridimensional mas suas restrições operacionais permitem transformá-lo em um PCPP. Por outro lado, as instâncias do PEUC são maiores que as do PCPP.

Os resultados computacionais do PCPP mostram que a LagClus e o AGC resolvem instâncias do PCPP que são consideradas difíceis para uma relaxação lagrangeana. No caso do PEUC, resultados computacionais mostram que a LagClus apresenta resultados (planos de estivagem) melhores que os praticados nos portos brasileiros.

O capítulo inicia com todos os conceitos, particularidades e resultados obtidos da LagClus e AGC para o PCPP. Em seguida, é mostrado o PEUC e os resultados da LagClus.

5.1 Introdução sobre o PCPP

O Problema do Carregamento de Paletes (PCP) consiste em: dado um conjunto de itens idênticos (caixas idênticas) com dimensões retangulares e um palete também retangular; procura-se otimizar a superfície do palete posicionando o máximo possível de caixas sobre o mesmo, sendo que as caixas não podem se sobrepor e todas devem estar posicionadas (empacotadas) sobre o palete, não podendo ultrapassar os limites de comprimento e largura do mesmo; elas ainda podem ser rotacionadas em 90° graus devendo estar com suas arestas paralelas às do palete. De acordo com Dyckhoff (1990), esse problema pode ser classificado como sendo 2/B/O/C (bidimensional, seleção de itens, objeto único, itens iguais), e é um caso especial dos problemas de corte e empacotamento.

O PCP aparece freqüentemente na logística de distribuição de produtos, pois o número total de caixas empacotadas pode reduzir os custos logísticos dado que um pequeno incremento nesse número pode estar representando uma economia significativa para a empresa.

Na literatura, existem dois tipos de problemas associados ao PCP (Hodgson, 1982): o Problema do Carregamento de Paletes do Produtor (PCPP) e o Problema do Carregamento de Paletes do Distribuidor (PCPD). No primeiro caso, as caixas possuem as mesmas dimensões, enquanto no segundo, apresentam dimensões diferentes. Em ambos

os casos, as caixas são empacotadas em camadas horizontais. A Figura 5.1 apresenta um exemplo de como as caixas podem ser empacotadas conforme o tipo de problema.

Considera-se aqui o problema do produtor com empacotamento bidimensional, ou seja, dada uma altura h de uma camada, o problema consiste em colocar o máximo número de caixas de faces (l, w) , sobre a superfície (L, W) do palete. As faces das caixas podem ser colocadas nas duas posições horizontais possíveis em cada camada, isto é, (l, w) e (w, l) .

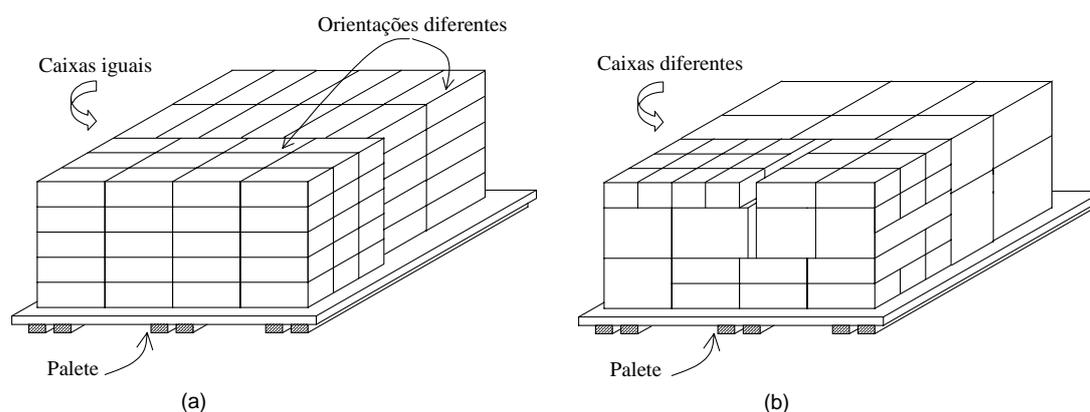


FIGURA 5.1 – Tipos de PCP: (a) Problema do Carregamento de Paletes do Produtor e (b) Problema do Carregamento de Paletes do Distribuidor (Adaptado de Morabito e Morales (1998)).

5.1.1 Revisão Bibliográfica para o PCPP

Considerando as diversas aplicações práticas do PCPP, muitos métodos de solução têm sido estudados. Boa parte dos algoritmos exatos disponíveis na literatura, utilizam, basicamente, uma estrutura em árvore (Dowsland, 1987b; Bhattacharya et al., 1998; Alvarez-Valdes et al., 2005a). Devido às dificuldades existentes, vários outros métodos foram criados ou utilizados, entre eles estão os métodos construtivos que dividem o palete em blocos (Steudel, 1979; Smith e De Cani, 1980; Young-Gun e Maing-Kyu, 2001), métodos recursivos (Morabito e Morales, 1998) e métodos baseados em estruturas do tipo G4 (Scheithauer e Terno, 1996) e do tipo L (Lins et al., 2003; Birgin et al., 2005). Existem outros trabalhos que aplicam as metaheurísticas conhecidas como Busca Tabu (Pureza e Morabito, 2006; Alvarez-Valdes et al., 2005b) e Algoritmos Genéticos (Herbert e Dowsland, 1996). Normalmente estas heurísticas utilizam como solução inicial um *layout* fornecido por uma heurística de blocos.

As heurísticas de blocos trabalham com subdivisões das dimensões do palete em função das dimensões das caixas. As possíveis combinações de unidades, com comprimento l e

largura w , na dimensão L ou W , que não excedem o comprimento ou a largura do palete, são definidas como possíveis partições (Bischoff e Dowsland, 1982).

Dado que (n, m) é um par ordenado inteiro positivo que satisfaz:

$$n \times l - m \times w \leq S \quad (5.1)$$

onde:

- S é uma dimensão do palete (L ou W);
- n é a quantidade de caixas com comprimento l na dimensão S ; e
- m é a quantidade de caixas com largura w na dimensão S .

Todas as possíveis partições da dimensão S são denotadas por $\mathfrak{S}(S, l, w)$. Se n e m também satisfazem:

$$0 \leq S - n \times l - m \times w < w \quad (5.2)$$

Então (n, m) é denominada de partição eficiente de S (Bischoff e Dowsland, 1982; Martins, 2003; Dowsland, 1987a). Para a dimensão S do palete, o conjunto de partições eficientes de S , denotado por $\Xi(S, l, w)$, pode ser definido usando a expressão $n \in \{0, 1, \dots, \lfloor S/l \rfloor\}$ e $m = \lfloor (S - n \times l)/w \rfloor$.

Com isso, uma partição eficiente constitui-se em colocar a maior quantidade de caixas, horizontalmente (comprimento) e verticalmente (largura), na dimensão L ou W , de forma a ocupar toda a dimensão, não restando espaço vazio, e se não for possível, o espaço restante deve ser menor que w .

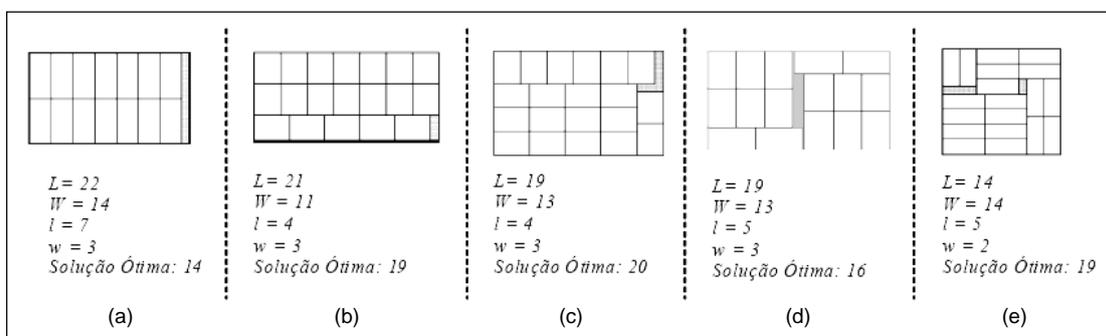


FIGURA 5.2 – Heurística de Blocos. (a) 1-Bloco, (b) 2-Blocos, (c) 3-Blocos, (d) 4-Blocos e (e) 5-Blocos. (Adaptado de Martins (2003)).

Uma vez definido o conceito de partições eficientes, as heurísticas de bloco consistem

em definir blocos (conjuntos) de caixas que deverão ser posicionadas sobre o palete. A Figura 5.2 apresenta várias soluções com 1, 2, 3, 4 e 5 blocos.

Considerando a heurística de cinco blocos de Smith e De Cani (1980), Bischoff e Dowsland (1982) propuseram um algoritmo usando quatro loops aninhados, baseados nas partições eficientes, para computar as dimensões de cinco blocos. Sendo assim, seja (L_i, W_i) as dimensões de um bloco i , e $C_i = L_i \times W_i$ a área desse bloco para todo $i = 1, \dots, 5$. Com isso, a Figura 5.3 apresenta o pseudocódigo da heurística de cinco blocos, conforme Bischoff e Dowsland (1982).

```

Pseudocódigo da Heurística de 5 Blocos

1  P_x ← {}
2  P_y ← {}
3  melhor_solução ← 0
4  P_x ← partições_eficientes_X
5  P_y ← partições_eficientes_y
6  Para cada P_x[i] Faça
7     n1 ← P_x[i].n
8     m1 ← P_x[i].m
9     L1 ← n1 * w
10    L2 ← m1 * l
11    Para cada P_y[j] Faça
12       n2 ← P_y[j].n
13       m2 ← P_y[j].m
14       W1 ← m2 * l
15       W4 ← n2 * w
16       C1 ← L2 * W1
17       Para cada P_x[z] Faça
18          n3 ← P_x[z].n
19          m3 ← P_x[z].m
20          L3 ← n3 * w
21          L4 ← m3 * l
22          C4 ← L4 * W4
23          Para cada P_y[k] Faça
24             n4 ← P_y[k].n
25             m4 ← P_y[k].m
26             W3 ← m4 * l
27             W2 ← n4 * w
28             C3 ← L3 * W3
29             C2 ← L2 * W2
30             Se Não_Existesobreposição (L1, L2, L3, L4, L5, W1, W2, W3, W4, W5) Então
31                Heurística_de_Um_Bloco (L5, W5, C5)
32                Se ((C1 + C2 + C3 + C4 + C5) > melhor_solução) Então
33                   Melhor_solução ← C1 + C2 + C3 + C4 + C5
34                   Salvar(L1, L2, L3, L4, L5, W1, W2, W3, W4, W5)
35             Fim se
36          Fim se
37       Fim para
38    Fim para
39 Fim para
40 Fim para

//Função que verifica se existe o quinto bloco
1  Não_Existesobreposição (L1, L2, L3, L4,
2     L5, W1, W2, W3, W4, W5)
3  Não_Existesobreposição ← Verdadeiro
4  Se (L1 + L3 > X) Então
5     Se (W1 + W3 > Y) Então
6        Não_Existesobreposição ← Falso
7     Senão
8        W5 ← Y - (W1 + W3)
9        L5 ← X - (L2 + L4)
10    Senão
11       Se (L2 + L4 > X) Então
12          Se (W2 + W4 > Y) Então
13             Não_Existesobreposição ← Falso
14          Senão
15             W5 ← Y - (W2 + W4)
16             L5 ← X - (L1 + L3)
17       Fim se
18    Fim se
19 Fim Não_Existesobreposição

```

FIGURA 5.3 – Pseudocódigo da Heurística de 5 Blocos.

Na literatura existem também limitantes superiores que consideram a geometria do PCPP e que são utilizados para avaliar a qualidade de uma solução dada por uma heurística lagrangeana, heurística ou metaheurística. Letchford e Amaral

(2001) apresentaram uma boa revisão bibliográfica sobre esses limitantes e realizaram experimentos para avaliar quais limitantes são dominantes no PCPP. Eles compararam os limitantes de Barnes (1979), o de Isermann (1987) e o limitante de empacotamento que consiste na relaxação de programação linear da formulação proposta por Beasley (1985). Os resultados mostraram que o limitante de programação linear domina todos os outros.

Esse problema analisado sob outro ponto de vista, é o clássico Problema do Máximo Conjunto Independente de Vértices (PMCIIV) (Dowland, 1987b). Sendo assim, o PCPP pode ser representado através de um grafo de conflitos em que cada vértice indica a localização do canto inferior esquerdo de uma caixa posta na horizontal ou vertical, e as arestas do grafo, as possíveis sobreposições das caixas.

Ao analisar o PPCP como um PMCIIV, pode-se imaginar que o arranjo das caixas próximas do canto inferior esquerdo do palete, não apresenta influência “direta” no arranjo das caixas próximas do canto superior direito do palete. Esse fato indica que existem *clusters* de vértices próximos dos cantos e, conseqüentemente, a LagClus pode ser aplicada.

5.1.2 Formulação do PPCP

Como mencionado por vários trabalhos na literatura (Morabito e Morales, 1998; Morabito e Farago, 2002; Alvarez-Valdes et al., 2005a), o PCPP pode ser formulado usando o caso particular da formulação de Beasley (1985) para o problema de corte não-guilhotinado bidimensional.

5.1.2.1 Definição dos *Normal Sets*

Considerando a Figura 5.4(a), seja L e W o comprimento e a largura do palete, respectivamente, tal que $L \geq W$, e, l e w o comprimento e a largura das caixas, respectivamente, tal que $l \geq w$ e $l \leq W$. Para representar os possíveis modos de empacotar uma caixa, seja $(l_1, w_1) = (l, w)$ e $(l_2, w_2) = (w, l)$, conforme mostra a Figura 5.4(b). Com isso, essas posições podem ser representadas por $(l_i, w_i)_{i=1,2}$, que indicam o comprimento e a largura de uma face na orientação i .

Para representar as posições das caixas no palete, sejam X e Y dois conjuntos que juntos são utilizados para definir as coordenadas (p, q) do canto inferior esquerdo das caixas. Esses conjuntos podem ser descritos como:

$$X = \left\{ p \in Z^+ \mid p = \sum_{i=1}^2 l_i b_i, 0 \leq p \leq L - w, b_i \geq 0 \text{ e inteiro } i = 1, 2 \right\} \quad (5.3)$$

$$Y = \left\{ q \in Z^+ \mid q = \sum_{i=1}^2 w_i b_i, 0 \leq q \leq W - w, b_i \geq 0 \text{ e inteiro } i = 1, 2 \right\} \quad (5.4)$$

Os conjuntos acima foram introduzidos por Christofides e Whitlock (1977) e são conhecidos como *normal sets*. A restrição das posições das caixas nas posições definidas por X e Y não restringem a solução ótima do problema.

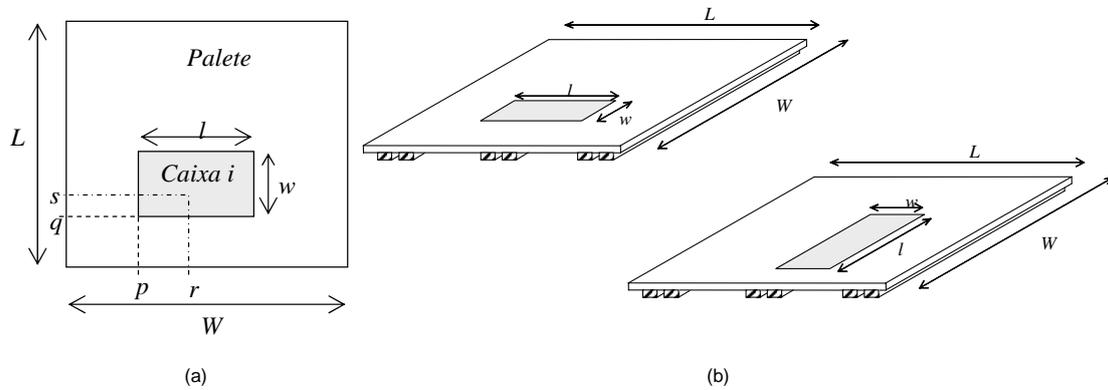


FIGURA 5.4 – Posição (r, s) não permitida em função da colocação de uma caixa na posição (p, q) com orientação i e (b) possíveis orientações para uma caixa na modelagem do PCPP.

5.1.2.2 Função de Incompatibilidade entre Posições no Pallet e o Modelo Matemático

Seja a uma função que descreve as restrições de sobreposições no pallet. Esta função pode ser obtida com antecedência para cada vértice (p, q) em relação a qualquer outro vértice (r, s) , para cada orientação i , sendo $p \in X \mid p \leq L - l_i$, $q \in Y \mid q \leq W - w_i$, $r \in X$, $s \in Y$, e $i = 1, 2$. Veja Figura 5.4 (a).

Assim, função a pode ser expressa como:

$$a_{i,p,q,r,s} = \begin{cases} 1, & \text{Se } 0 \leq p \leq r \leq p + l_i - 1 \leq L - 1 \text{ e } 0 \leq q \leq s \leq q + w_i - 1 \leq W - 1 \\ 0, & \text{Caso Contrário} \end{cases} \quad (5.5)$$

Agora, seja $x_{i,p,q} \in \{0, 1\}$ uma variável binária de decisão para todo $p \in X \mid p \leq L - l_i$,

$q \in Y | q \leq W - w_i$, e $i = 1, 2$. Se $x_{i,p,q} = 1$, uma caixa é colocada nas coordenadas (p, q) do paleta com a orientação i , caso contrário, $x_{i,p,q} = 0$.

Com isso, o PCPP pode ser formulado como (Beasley, 1985):

$$v(PCPP) = Max \left\{ \sum_{i=1}^2 \sum_{\{p \in X | p \leq L - l_i\}} \sum_{\{q \in Y | q \leq W - w_i\}} x_{i,p,q} \right\} \quad (5.6)$$

(PCPP) Sujeito a:

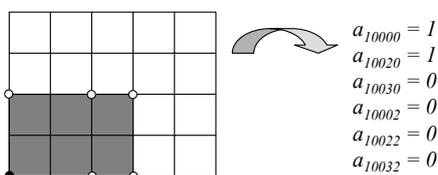
$$\sum_{i=1}^2 \sum_{\{p \in X | p \leq L - l_i\}} \sum_{\{q \in Y | q \leq W - w_i\}} a_{i,p,q,r,s} x_{i,p,q} \leq 1, \quad \forall r \in X \text{ e } s \in Y \quad (5.7)$$

$$x_{i,p,q} \in \{0, 1\} \quad \forall i = 1 \dots 2, p \in X \text{ tal que } p \leq L - l_i, \text{ e } q \in Y \text{ tal que } q \leq W - w_i \quad (5.8)$$

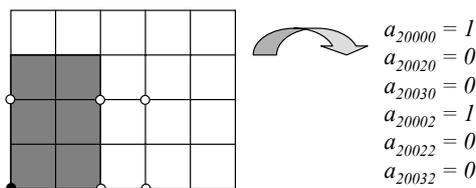
As restrições 5.7 garantem a não existência de sobreposição de caixas e as definidas em 5.8 que as variáveis de decisão são binárias.

A seguir será apresentado um exemplo da formulação do PCPP adaptado de Oliveira (2005). Considere que as dimensões do paleta são $(L, W) = (5, 4)$ e as dimensões das caixas $(l, w) = (3, 2)$. O canto inferior esquerdo de uma caixa qualquer pode ser posicionado um ponto dado pelos conjuntos $X = \{0, 2, 3\}$ e $Y = \{0, 2\}$ conforme as Equações 5.3 e 5.4, respectivamente. Definindo agora os valores da função a :

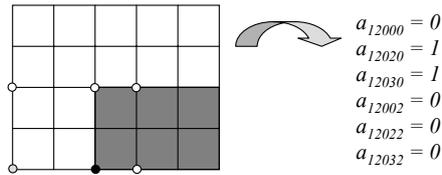
- Para $(p, q) = (0, 0)$ e $i = 1$:



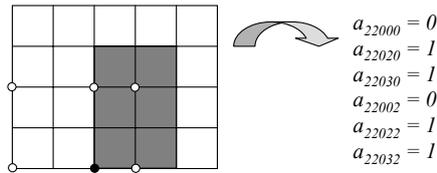
- Para $(p, q) = (0, 0)$ e $i = 2$:



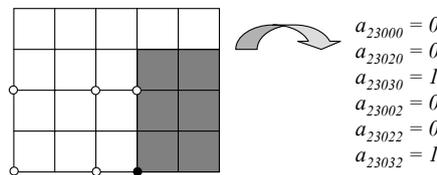
- Para $(p, q) = (2, 0)$ e $i = 1$:



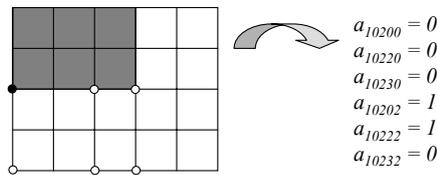
- Para $(p, q) = (2, 0)$ e $i = 2$:



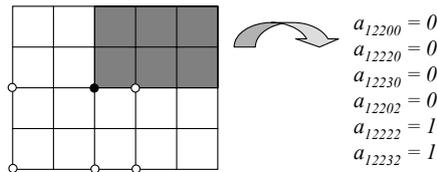
- Para $(p, q) = (3, 0)$ e $i = 2$:



- Para $(p, q) = (0, 2)$ e $i = 1$:



- Para $(p, q) = (2, 2)$ e $i = 1$:



Agora as restrições definidas em 5.7 podem ser escritas como:

- Para $(r, s) = (0, 0)$: $x_{100} + x_{200} \leq 1$;
- Para $(r, s) = (0, 2)$: $x_{102} + x_{200} \leq 1$;
- Para $(r, s) = (2, 0)$: $x_{100} + x_{120} + x_{220} \leq 1$;
- Para $(r, s) = (2, 2)$: $x_{102} + x_{122} + x_{220} \leq 1$;
- Para $(r, s) = (3, 0)$: $x_{120} + x_{220} + x_{230} \leq 1$;

- Para $(r, s) = (3, 2)$: $x_{122} + x_{220} + x_{230} \leq 1$.

Assim, o problema com $(L, W) = (5, 4)$ e $(l, w) = (3, 2)$ pode ser formulado da seguinte maneira:

$$v(PCPP) = \text{Max}(x_{100} + x_{102} + x_{120} + x_{122} + x_{200} + x_{220} + x_{230})$$

(PCPP) Sujeito a:

$$x_{100} + x_{200} \leq 1$$

$$x_{102} + x_{200} \leq 1$$

$$x_{100} + x_{120} + x_{220} \leq 1$$

$$x_{102} + x_{122} + x_{220} \leq 1$$

$$x_{120} + x_{220} + x_{230} \leq 1$$

$$x_{122} + x_{220} + x_{230} \leq 1$$

$$x_{100}, x_{102}, x_{120}, x_{122}, x_{200}, x_{220}, x_{230} \in \{0, 1\}$$

Para esse problema, a solução ótima é dada por $x_{100} = x_{102} = x_{230} = 1$ e $x_{120} = x_{122} = x_{200} = x_{220} = 0$ com o valor da função objetivo igual a 3. A Figura 5.5 mostra graficamente a distribuição ótima das caixas sobre o palete.

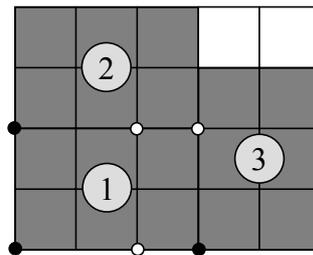


FIGURA 5.5 – Solução ótima para o problema $(L, W) = (5, 4)$ e $(l, w) = (3, 2)$.

5.1.2.3 Comparação entre o PPCP e o PMCIIV

Conforme citado anteriormente, o PCPP pode ser formulado como um PMCIIV, porém o PMCIIV produz mais restrições que a formulação definida em 5.6-5.8. Isso decorre do fato de que a formulação de Beasley (1985) explora o uso das cliques, reduzindo assim o número de restrições.

Por exemplo, considere novamente um paleta com dimensões $(L, W) = (5, 4)$ e caixas

$(l, w) = (3, 2)$ (igual ao do exemplo da Seção 5.1.2.2).

Agora, usando a abordagem dada pelo PMCIV, a Figura 5.6(b) mostra o grafo de conflitos obtido para o problema, e em (c) é apresentada a formulação do PMCIV. Como indicado antes, o PMCIV gera mais restrições que a modelagem de Beasley (1985), entretanto elas estão consideradas implicitamente nas cliques presentes na formulação do PCPP.

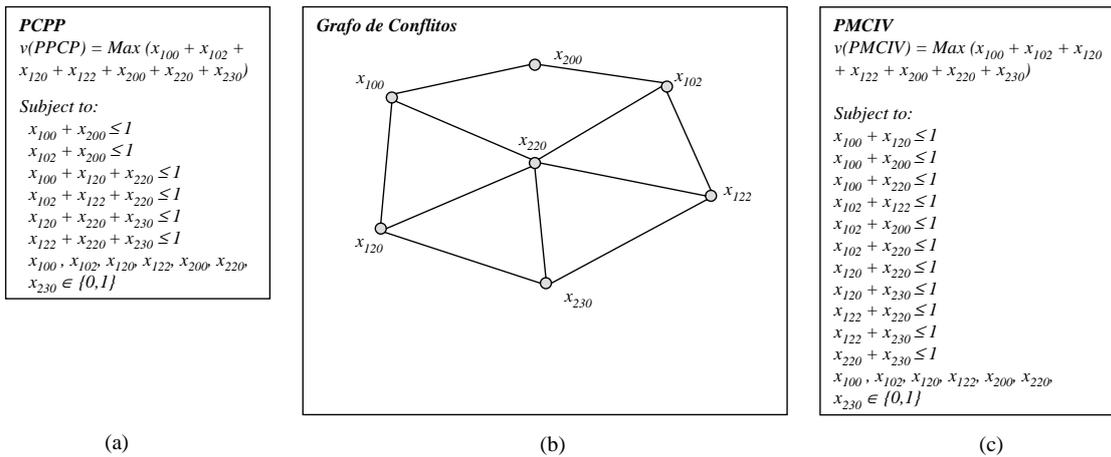


FIGURA 5.6 – Comparação entre a formulação de Beasley (1985) e a do PMCIV. (a) Formulação de Beasley (1985), (b) grafo de conflitos e (c) PMCIV.

5.1.3 LagClus para o PCPP

Dada a semelhança entre o PCPP e o PMCIV, e considerando a formulação de Beasley (1985), um grafo de conflitos $G = (V, A)$ pode ser obtido conforme a Figura 5.6 (b). Assim, a LagClus pode ser aplicada no PCPP seguindo os passos básicos descritos no Capítulo 3 para o PMCIV:

- a) Aplique uma heurística de particionamento para dividir G em \bar{P} partes, formando \bar{P} clusters;
- b) Relaxe as restrições presentes no PCPP que possuem vértices em diferentes clusters. Em cada clique relaxada k , verifique se existem pares de vértices, ou até mesmo cliques dominadas por k , que possuem vértices pertencentes ao mesmo cluster. Se existirem, as adicione nos respectivos clusters, ou seja, adicione uma restrição de adjacência para cada par de vértices encontrado, ou as restrições de clique dominadas com vértices pertencentes ao mesmo cluster; e
- c) A relaxação lagrangeana é decomposta em \bar{P} subproblemas e resolvida.

Deve-se observar que no segundo passo se alguma clique for relaxada, esta deve ser decomposta e cada uma de suas arestas analisada. Se uma aresta conecta dois vértices de um mesmo *cluster*, a mesma deve ser acrescentada no respectivo *cluster*. No caso de existirem sub cliques dominadas em que todos os vértices pertencem a um mesmo *cluster*, estas devem ser introduzidas nos respectivos *clusters*.

Esse procedimento fortalece a relaxação lagrangeana e evita soluções inválidas nos *clusters*. A Figura 5.7 apresenta um exemplo sobre essa questão. Note que se a restrição de adjacência $x_1 + x_2 \leq 1$ não for adicionada no *Cluster 1* e $x_3 + x_4 \leq 1$ no *Cluster 2*, ambos os *clusters* poderiam gerar soluções inválidas, isto é, poderiam gerar a seguinte solução $x^{Cluster_1} = \{1, 1\}$ e $x^{Cluster_2} = \{1, 1\}$.

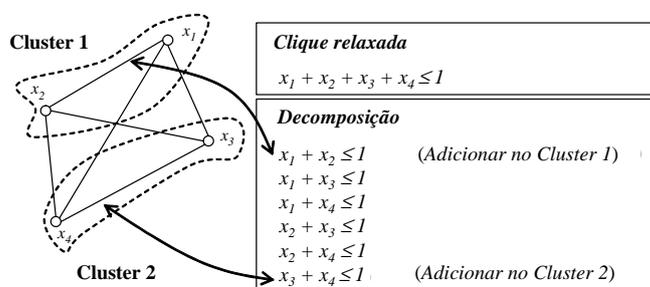


FIGURA 5.7 – Exemplo de como decompor uma clique relaxada nos *clusters*.

5.1.3.1 Formulação do PCPP com a Divisão do Grafo de Conflitos

Considerando a decomposição do grafo de conflitos do PCPP em \bar{P} *clusters*, tem-se a seguinte formulação em bloco do PCPP:

$$v(PCPP) = \text{Max} \sum_{p=1}^{\bar{P}} x^p \quad (5.9)$$

(PCPP) Sujeito a:

$$+A^1 x^1 + A^2 x^2 \cdots + A^{\bar{P}} x^{\bar{P}} \leq 1 \quad (5.10)$$

$$\begin{aligned} +D^1 x^1 &\leq 1 \\ &+D^2 x^2 &\leq 1 \\ &\cdots &\leq \vdots \\ &+D^{\bar{P}} x^{\bar{P}} &\leq 1 \end{aligned} \quad (5.11)$$

$$x^1 \in B^{n_1} \quad \dots \quad \dots \quad x^P \in B^{n_P} \quad (5.12)$$

Sendo:

- A^p uma matriz binária de dimensão $M \times |V|$ que representa os coeficientes associados às variáveis x^p do *cluster* p que aparecem nas M restrições de clique relaxadas;
- D^p uma matriz binária de dimensão $K - M \times |V|$ que representa os coeficientes associados às variáveis x^p do *cluster* p que aparecem nas cliques do *cluster* p , sendo K o número total de restrições de clique definida em 5.7; e
- B^{n_p} um vetor de variáveis binárias com dimensão n_p associadas às variáveis x^p .

Note que o problema 5.9-5.12 tem a mesma estrutura do modelo 3.3-3.6 descrito para o PMCIV. A diferença está no tipo de restrições relaxadas. No PMCIV, as restrições de adjacência é que são relaxadas, enquanto no PCPP, são as cliques do modelo de Beasley (1985).

Sendo assim, as formulações definidas pelas equações 3.8 para os subproblemas, e 3.9 para a LagClus, podem ser aqui consideradas.

5.1.3.2 Heurística Lagrangeana

A heurística lagrangeana implementada para ser utilizada durante o algoritmo de subgradientes, denominada aqui de Heurística de Verificação e Melhoria (HVM), primeiramente identifica todos os vértices da solução relaxada que possuem vértices conflitantes, em seguida o vértice com o maior número de vértices em conflito é removido da solução relaxada. O processo é então novamente repetido até que uma solução sem conflitos seja encontrada. Em seguida, essa heurística busca inserir novos vértices, não removidos previamente, na solução viável encontrada procurando obter uma solução viável melhor.

A Figura 5.8 apresenta a HVM. O tamanho do passo do algoritmo de subgradiente é atualizado considerando os limitantes lagrangeanos fornecidos pela LagClus e as soluções viáveis obtidas pela HVM. Além disso, antes da primeira iteração do algoritmo de subgradientes, a heurística de blocos proposta por Smith e De Cani (1980) é utilizada para gerar uma solução inicial viável para o problema.

5.1.4 AGC para o PCPP usando a Idéia dos *Clusters*

Usando o particionamento mostrado na Seção 5.1.3.1 descrita anteriormente, a decomposição Dantzig-Wolfe pode ser aplicada como mostrado no Capítulo 3, dado que o PCPP é um PMCIIV:

$$v(PPCP_{DW})_{PL} = Max \left\{ \sum_{p=1}^{\bar{P}} \sum_{j \in J_p} w^{j,p} \lambda_{j,p} \right\} \quad (5.13)$$

(PPCP_{DW})_{PL} Sujeito a:

$$\sum_{p=1}^{\bar{P}} \sum_{j \in J_p} \lambda_{j,p} [A_p] x^{j,p} \leq 1 \quad (5.14)$$

$$\sum_{j \in J_p} \lambda_{j,p} = 1 \quad \forall p \in \{1, \dots, \bar{P}\} \quad (5.15)$$

$$\lambda_{j,p} \geq 0 \quad \forall p \in \{1, \dots, \bar{P}\} \text{ e } j \in J_p \quad (5.16)$$

Sendo:

- $\lambda_{j,p}$ uma variável de decisão que representa o ponto extremo $j \in J_p$;
- J_p o conjunto de pontos extremos do *cluster* p ; e
- $w^{j,p}$ o custo da solução x^j do subproblema p .

Heurística de Verificação e Melhoria - HVM

1. Faça o vetor de solução factível (x_f) igual ao vetor da solução relaxada dado pela LagClus;
2. **Enquanto** não obtiver uma solução factível **Faça**
 3. Para cada vértice i de x_f , defina o número de vértices j conflitantes com i ;
 4. Ordene, decrescentemente, x_f conforme o número de vértices conflitantes;
 5. Elimine o primeiro vértice presente em x_f ;
6. **Fim** do enquanto;
7. Testar entre os demais vértices não presentes em x_f e não presentes naquele conjunto de vértices removido no passo 2, se é possível inserir mais algum em x_f .

FIGURA 5.8 – Heurística de Verificação e Melhoria (HVM) utilizada no algoritmo de subgradientes para o PCPP.

Observe que o modelo 5.13-5.16 é o mesmo descrito no Capítulo 3 (modelo 3.20-3.23), porém aqui as restrições descritas 5.14 são cliques conforme a formulação de Beasley (1985).

Novamente conforme mostrado no Capítulo 3, as equações 3.24 e 3.25 podem ser utilizadas para realizar a equivalência entre LagClus e AGC.

5.1.4.1 O Conjunto Inicial de Colunas

Para gerar o conjunto inicial de colunas do AGC, realizou-se uma adaptação da heurística RSF (*Recursive-Smalest-First*), proposta por Yamamoto e Lorena (2005) para o PRCP modelado como um PMCIV. A heurística RSF inicia escolhendo um vértice x de grau mínimo, em seguida, ela torna o vértice escolhido e seus vértices adjacentes inativos. A partir da lista de vértices ativos, é calculado novamente o grau de cada vértice, e em seguida, um novo vértice de grau mínimo é selecionado, e o ciclo é então repetido. O algoritmo termina quando não restarem mais vértices ativos. Os vértices escolhidos formam um conjunto independente de vértices, no caso do PCPP um *layout* das caixas sobre o palete (uma solução).

A RSF foi usada da seguinte maneira para gerar o conjunto inicial de colunas do PCPP. Primeiro, ao invés de escolher o vértice de grau mínimo, escolhe-se aleatoriamente um vértice x . Em seguida, o vértice escolhido e seus vértices adjacentes são colocados inativos. Daí em diante, o processo é idêntico ao da heurística original, ou seja, calcula-se o grau de cada um dos vértices restantes e aquele de menor grau é escolhido, novamente o vértice escolhido e seus vértices adjacentes são colocados inativos, e o processo é repetido. Quando a heurística termina, os vértices selecionados formam uma solução para o PCPP. Todo esse algoritmo é repetido até que se obtenha o número desejado de soluções iniciais. Sendo ND esse número, o número de colunas adicionadas no PMR é igual a $ND \times \bar{P}$, pois uma solução x é dividida em \bar{P} *clusters*.

Para dar uma idéia melhor do AGC aqui utilizado, a Figura 5.9 apresenta um diagrama com o fluxo e os passos principais utilizados. Note que a área mais escura representa o processo de geração de colunas.

O AGC apresentado na Figura 5.9 termina quando nenhuma outra coluna de custo reduzido positivo é gerada.

5.1.4.2 Diversidade do Conjunto Inicial de Colunas

A aleatoriedade aplicada na RSF para gerar o conjunto inicial de colunas, permite que vértices diferentes apareçam em soluções viáveis do PCPP. Isso indica que, possivelmente, soluções parcialmente boas possam ser obtidas para cada um dos *clusters* (lembre-se que uma vez selecionada um vértice para entrar na solução, ele e os seus vértices adjacentes são colocados como inativos).

Essa diversidade pode ser notada no AGC uma vez que, como indicado no Capítulo 3, a combinação de colunas boas de cada um dos *clusters*, permite obter uma solução boa para o problema primal.

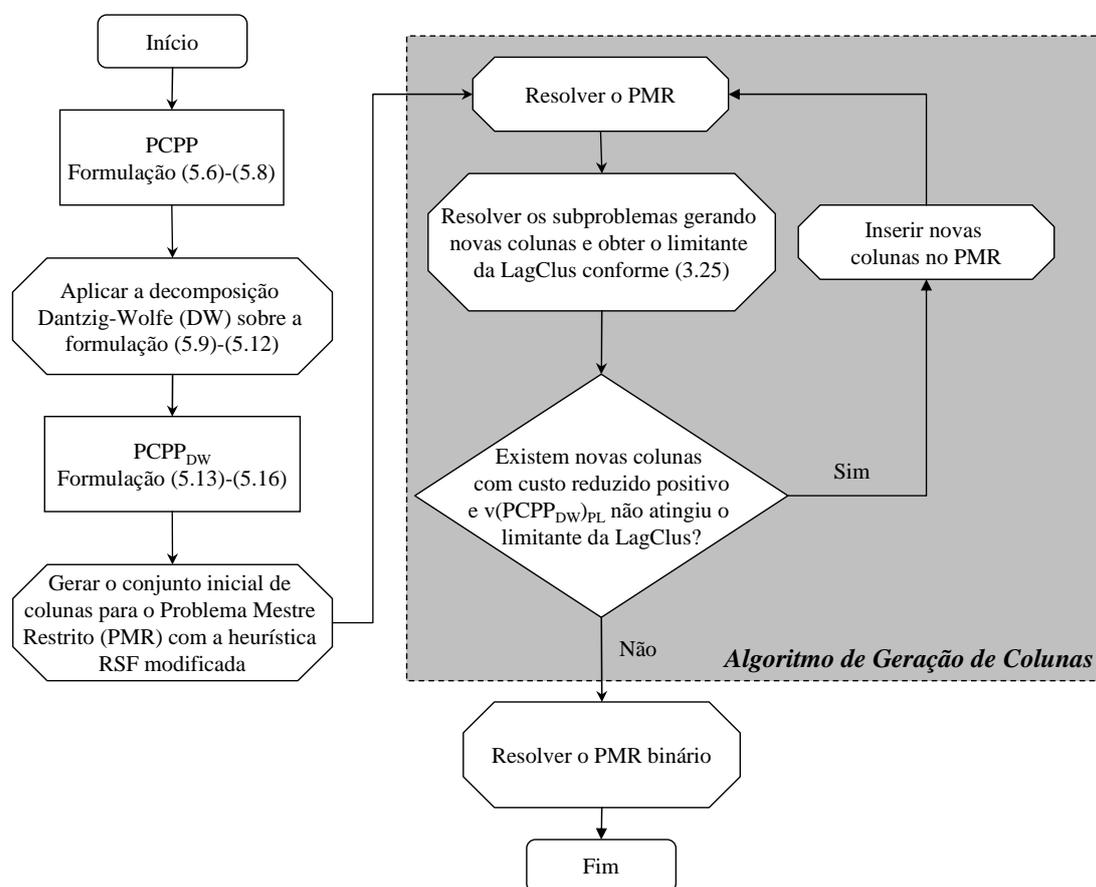


FIGURA 5.9 – Diagrama dos passos da AGC para o PCPP.

5.1.5 Resultados Computacionais

Na literatura, existem vários trabalhos que relatam instâncias testes como em Dowsland (1987b), Letchford e Amaral (2001), Morabito e Morales (1998), Alvarez-Valdes et al. (2005a) e Pureza e Morabito (2006). Outros trabalhos relatam instâncias testes obtidas de

problemas reais junto a transportadoras como em Morabito et al. (2000) e em Morabito e Farago (2002).

Neste trabalho são consideradas algumas dessas instâncias propostas divididas em três grupos. O primeiro grupo é composto pelas 10 instâncias (L1-L10) propostas por Letchford e Amaral (2001). Na literatura, essas instâncias são consideradas complicadas para uma relaxação lagrangeana e para a relaxação de programação linear (veja Letchford e Amaral (2001)). O segundo grupo é composto por 10 instâncias (L11-L20) aleatoriamente obtidas do conjunto COVER II proposto por Dowsland (1987b), e por último, o terceiro grupo também é composto de 10 instâncias (L21-L30) obtidas também aleatoriamente do conjunto COVER III proposto por Alvarez-Valdes et al. (2005a). Essas últimas instâncias não apresentam solução ótima conhecida.

A codificação foi feita em C++ e os testes em um Pentium IV 2,66 GHz com 512 MB de memória RAM. O CPLEX 7.5 foi utilizado para resolver os subproblemas tanto da LagClus quanto do AGC, bem como o PMR. O particionamento foi novamente realizado utilizando o METIS.

Como no PRCP, o algoritmo de subgradientes descrito por Narciso e Lorena (1999), foi aqui implementado para resolver o dual lagrangeano. Em uma iteração k , os subgradientes são definidos como $g^{\lambda_k} = 1 - \sum_{p=1}^{\bar{P}} A^p x^{\lambda_k}$ tal que x^{λ_k} é a solução ótima da LagClus ($LG_{\lambda}PPCP$).

O algoritmo de subgradientes atualiza os multiplicadores lagrangeanos fazendo $\lambda_{k+1} = \lambda_k + \theta_k g^{\lambda_k}$ onde θ_k é o tamanho do passo calculado como $\theta_k = \frac{\pi(ub_k - lb_k)}{\|g^{\lambda_k}\|^2}$, sendo lb_k a melhor solução factível encontrada até a iteração k , e ub_k o melhor limitante dual encontrado também até a iteração k . O controle utilizado para o parâmetro π foi o mesmo utilizado no PRCP. Inicia-se com o valor 2 e se durante 15 iterações o valor do limite superior (ub_k) não decrescer, π é dividido pela metade. O algoritmo de subgradientes termina quando:

- a) $\pi \leq 0,005$;
- b) $(ub_k - lb_k) < 1$; e
- c) $\|g^{\lambda_k}\|^2 = 0$.

5.1.6 Resultados da LagClus

As Tabelas 5.1, 5.2 e 5.3 apresentam os resultados da LagClus para o PCPP, sendo que as colunas referem-se a:

- Inst. - Nome dado à instância;
- L e W - Comprimento e largura do palete, respectivamente;
- l e w - Comprimento e largura das caixas, respectivamente;
- v^* - Solução ótima do problema;
- v_{Melhor} - Melhor solução conhecida reportada na literatura;
- UB_{Area} - Limitante de área dado por $\lfloor (L \times W)/(l \times w) \rfloor$;
- UB_{Barnes} - Limitante de Barnes (1979);
- UB_{PL} - Limitante de PL;
- $Tempo_1$ (s) - Tempo em segundos utilizado pelo CPLEX 7.5 para resolver a relaxação de PL;
- LB - Limitante inferior fornecido pela HVM;
- UB - Limitante superior fornecido pela LagClus;
- GAP LB (%) - *Gap* percentual obtido considerando a solução ótima ou a melhor conhecida em relação ao limitante inferior $Gap\ LB = \frac{(v^* \text{ ou } v_{Melhor} - LB)}{v^* \text{ ou } v_{Melhor}} * 100$;
- GAP UB (%) - *Gap* percentual obtido considerando a solução ótima ou a melhor conhecida em relação ao limitante superior $Gap\ UB = \frac{(UB - v^* \text{ ou } v_{Melhor})}{v^* \text{ ou } v_{Melhor}} * 100$;
- $Tempo_2$ (s) - Tempo em segundos utilizado pela LagClus até encontrar uma das condições de parada; e
- Iter. - Número de iterações utilizado pela LagClus até encontrar uma das condições de parada.

Como pode ser visto nas Tabelas 5.1, 5.2 e 5.3, os limitantes inferiores gerados pela heurística HVM estão muito próximos da solução ótima ou da melhor solução conhecida. No pior caso, os resultados diferem em uma caixa apenas. Os limitantes duais da LagClus e da PL podem ser considerados os mesmos, novamente no pior caso, e os seus tempos computacionais são comparáveis, veja Tabela 5.3.

TABELA 5.1 – Resultados computacionais para 10 instâncias propostas por Letchford e Amaral (2001), consideradas difíceis para uma relaxação lagrangeana.

Inst.	L	W	l	w	v^*	UB _{Area}	UB _{Barnes}	PL		LagClus					
								Tempo ₁		GAP		GAP		Tempo ₂	Iter.
								UB _{PL}	(s)	LB	UB	LB(%)	UB(%)		
L1	32	22	5	4	34	35	35	35,00	0,27	34	35,00	0	2,95	25	145
L2	32	27	5	4	42	43	43	43,00	0,26	42	43,00	0	2,39	88	145
L3	40	26	7	4	36	37	37	37,00	0,15	36	37,00	0	2,78	98	145
L4	40	33	7	4	46	47	47	47,00	0,82	46	47,00	0	2,18	276	145
L5	53	26	7	4	48	49	49	49,00	0,59	48	49,01	0	2,11	318	145
L6	37	30	8	3	45	46	46	46,00	0,73	45	46,03	0	2,30	202	145
L7	81	39	9	7	49	50	50	50,00	1,34	49	49,99	0	2,03	257	67
L8	100	64	17	10	36	37	37	37,00	0,25	36	37,00	0	2,78	114	145
L9	100	82	22	8	45	46	46	46,00	1,90	45	46,03	0	2,28	404	145
L10	100	83	22	8	45	47	46	46,00	1,86	45	46,03	0	2,28	403	145

A Tabela 5.1 apresenta todos os resultados para as instâncias L1-L10 com 2 *clusters*. Apesar dessas instâncias serem consideradas difíceis para uma relaxação lagrangeana, a LagClus foi capaz de provar a solução ótima da instância L7 e para as outras, os limitantes duais ficarem próximos da solução ótima (aproximadamente 1 caixa).

A Tabela 5.2 apresenta os resultados para o segundo grupo de instâncias. Nesse caso, foram utilizados 5 *clusters*. Para essas instâncias a LagClus foi capaz de garantir a solução ótima em 60% dos casos: L13, L14, L16, L18, L19 e L20. Novamente os limitantes duais ficaram próximos da solução ótima.

TABELA 5.2 – Resultados computacionais para as 10 instâncias aleatoriamente escolhidas da COVER II (Dowland, 1987b).

Inst.	L	W	l	w	v^*	UB _{Area}	UB _{Barnes}	PL		LagClus					
								Tempo ₁		GAP		GAP		Tempo ₂	Iter.
								UB _{PL}	(s)	LB	UB	LB(%)	UB(%)		
L11	57	53	7	5	85	86	86	86,14	68,09	85	86,36	0	1,6	418	145
L12	84	75	11	6	94	95	95	95,17	72,56	94	95,62	0	1,73	1192	145
L13	151	131	19	11	94	94	94	95,17	209,39	93	94,50	1,06	0,53	2755	145
L14	61	38	6	5	77	77	77	77,07	22,01	76	77,24	1,30	0,32	144	145
L15	100	53	9	7	83	84	84	83,90	55,14	83	84,12	0	1,35	365	145
L16	120	80	14	11	61	62	62	61,75	6,03	61	61,98	0	1,64	41	115
L17	51	38	11	3	57	58	58	57,75	5,73	57	58,36	0	2,39	118	145
L18	120	83	17	6	97	97	97	97,50	90,68	97	97,99	0	1,02	273	55
L19	131	86	16	7	100	100	100	100,14	230,52	100	100,99	0	0,99	195	35
L20	98	93	17	7	75	76	76	75,00	89,97	75	75,99	0	1,57	482	135

A Tabela 5.3 apresenta os resultados encontrados para as últimas 10 instâncias com 15 *clusters*. Diferentemente das Tabelas 5.1 e 5.2 em que a LagClus foi capaz de provar soluções ótimas, neste último conjunto de instâncias ela não encontrou nenhuma solução ótima.

TABELA 5.3 – Resultados computacionais para as 10 instâncias aleatoriamente escolhidas da COVER III (Alvarez-Valdes et al., 2005a), sobre todas aquelas que não apresentam solução ótima conhecida.

Inst.	L	W	l	w	v_{Melhor}	UB _{Area}	UB _{Barnes}	PL		LagClus					
								UB _{PL}	Tempo ₁ (s)	LB	UB	GAP LB(%)	GAP UB(%)	Tempo ₂ (s)	Iter.
L21	99	88	12	5	144	145	145	145,00	1370,07	144	145,83	0	1,27	998	145
L22	99	75	13	5	113	114	114	114,00	233,54	113	114,70	0	1,51	342	145
L23	97	95	9	7	145	146	146	146,14	1184,67	145	146,82	0	1,26	620	145
L24	98	98	10	7	136	137	137	137,13	1480,91	136	137,68	0	1,23	468	145
L25	98	88	10	7	122	123	123	123,12	594,73	122	123,66	0	1,36	357	145
L26	97	96	11	6	140	141	141	141,00	891,98	140	141,86	0	1,33	932	145
L27	96	87	8	7	148	149	149	149,00	1232,13	148	149,95	0	1,32	638	145
L28	99	70	15	4	114	115	115	115,00	520,66	114	116,17	0	1,90	348	145
L29	91	70	12	5	105	106	106	106,00	136,84	105	106,52	0	1,45	156	145
L30	93	84	11	6	117	118	118	118,00	248,46	117	118,92	0	1,64	387	145

Para mostrar a relação entre a qualidade do limitante e o tempo utilizado pela LagClus, um outro experimento variando o número de *clusters* foi realizado. A instância L7 foi utilizada como teste e os resultados estão mostrados na Tabela 5.4. Como esperado, na medida em que o número de *clusters* cresce, a qualidade do limitante da LagClus decresce bem como o tempo computacional.

TABELA 5.4 – Resultados computacionais da LagClus para a instância L7 variando o número de *clusters*.

# de Clusters	Iter.	LagClus		
		LB	UB	Tempo (s)
2	67	49	49,9936	257
3	145	49	50,0612	86
4	145	49	50,0429	31
5	145	49	50,1019	33
6	145	49	50,1316	27
7	145	49	50,1360	18
8	145	49	50,1512	18
9	145	49	50,1241	19
10	145	49	50,1229	22
11	145	49	50,1785	21
12	145	49	50,2020	21
13	145	49	50,2199	16
14	145	49	50,1731	18
15	145	49	50,2645	15

5.2 Resultados do AGC

Os resultados encontrados com o AGC para as instâncias L1-L10 estão mostrados na Tabela 5.5 em que as colunas representam:

- # Inicial de Colunas - Número inicial de colunas obtidas com a heurística RSF;
- PMR Inicial - Valor inicial de $v(PCPP_{DW})_{PL}$ com o conjunto inicial de colunas;
- # Final de Colunas - Número final de colunas no PMR;
- PMR Final - Valor final do $v(PCPP_{DW})_{PL}$ com todas as colunas;
- Tempo₁ (s) - Tempo em segundos utilizado pelo processo de geração de colunas;
- PMR Binário - Valor do $v(PCPP_{DW})$ com todas as variáveis binárias; e
- Tempo₂ (s) - Tempo em segundos utilizado pelo CPLEX 7.5 para resolver o PMR binário.

No AGC não foi utilizada nenhuma técnica de eliminação de colunas. Isso para verificar o seu comportamento no pior caso.

TABELA 5.5 – Resultados obtidos com a Geração de Colunas.

Inst.	AGC					Resolvendo o PMR final de forma inteira (PLI)	
	# Inicial de Colunas	PMR Inicial	# Final de Colunas	PMR Final	Tempo ₁ (s)	PMR Binário	Tempo ₂ (s)
L1	500	34,00	626	35,00	17	34	0,00
L2	500	41,00	674	43,00	45	42	0,20
L3	500	35,16	697	37,00	62	36	0,00
L4	500	45,00	680	47,00	183	46	2,00
L5	500	47,20	691	49,00	286	48	1,00
L6	500	44,00	829	46,00	194	45	0,00
L7	500	49,00	815	49,85	2166	49	1,00
L8	500	36,00	645	37,00	27	36	0,00
L9	500	44,00	813	46,00	331	45	4,01
L10	500	44,00	813	46,00	326	45	5,10

Como mostrado na Tabela 5.5, os resultados validam a decomposição utilizada neste trabalho, embora o tempo computacional tenha sido elevado para uma instância, compare com a Tabela 5.1. Mas vale ressaltar que não foi utilizada nenhuma técnica de eliminação de colunas improdutivas, ou parada prematura do processo de geração devido a algum

critério pré-estabelecido. Por outro lado, após gerar todas as colunas, o PMR obtido e resolvido de forma inteira, encontrou todas as soluções ótimas em um tempo razoável, inferior a 5,10s para todas as instâncias.

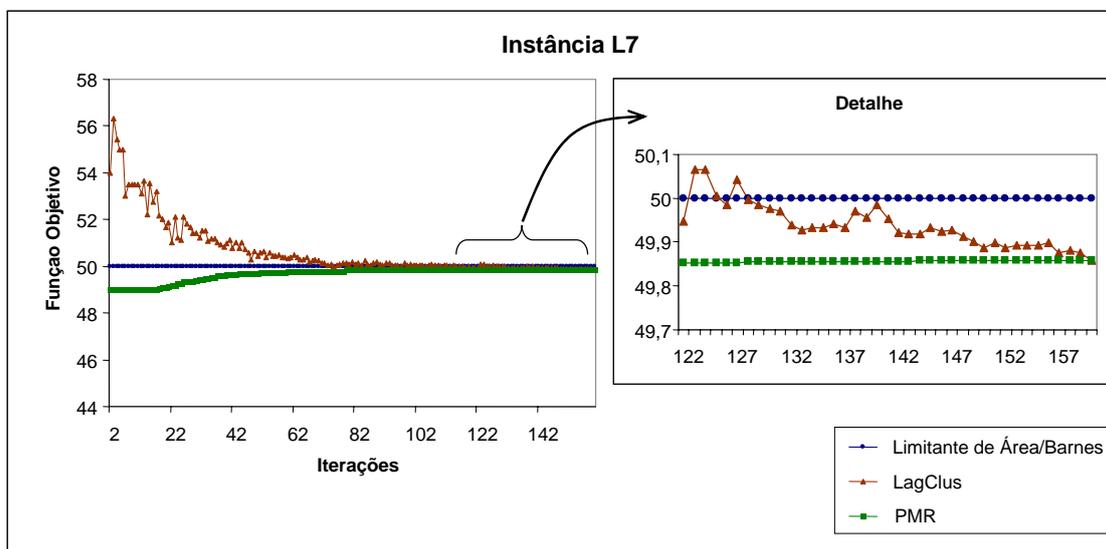


FIGURA 5.10 – Comportamento do problema mestre restrito e da LagClus obtida a partir do multiplicadores duais do PMR.

A Figura 5.10 mostra o comportamento do AGC e da LagClus obtida com os multiplicadores duais do PMR. Pode-se perceber que, conforme o número de iterações aumenta, os dois limitantes tendem a se igualar, muito próximo do limitante teórico de área dado por $\lfloor (L \times W) / (l \times w) \rfloor$. Percebe-se ainda que o AGC poderia ter parado antes, próximo da iteração de número 122, pois o limitante dado pela LagClus ultrapassou o limitante de área. Observações como essas podem reduzir o tempo computacional utilizado no processo.

A seguir é apresentado o PEUC e os resultados computacionais da LagClus sobre algumas instâncias reais desse problema.

5.3 Introdução sobre o PEUC

Esta seção considera o Problema da Estivagem de Unidades de Celulose (PEUC) em porões de navios dedicados para o transporte marítimo internacional. Segundo dados de 2007 da Associação Brasileira de Celulose e Papel - BRACELPA (<http://www.bracelpa.org.br>), o Brasil é o 7º maior produtor mundial de celulose de todos os tipos, ocupa a liderança na produção de celulose de fibra curta de mercado e a 11ª posição na produção de papel. Em 2006, as empresas brasileiras produziram 11,1 milhões de toneladas de celulose e 8,6 milhões de toneladas de papel, gerando uma participação de

1,4% no PIB.

A produção de celulose brasileira é obtida de duas espécies de madeira: o Eucalipto e o Pínus. A madeira, uma vez cortada e descascada, é transportada até a unidade industrial onde será processada. Ao final, placas de celulose são geradas, empilhadas e amarradas para compor uma unidade de celulose. As unidades são então transportadas por navios especializados para atingir o mercado externo. Porém, para que esse transporte seja realizado, as unidades de celulose devem ser estivadas nos porões dos navios.

A estivagem de cargas consiste em arranjar itens menores (unidades de carga) em uma unidade de transporte da melhor maneira possível procurando reduzir os custos logísticos envolvidos (Branch, 1996). O planejamento da estivagem deve considerar o tipo de carga a ser estivada, o tipo de unidade de transporte, a natureza da carga, dentre outros fatores (Handabaka, 1994).

O produto final da etapa do planejamento da estivagem é o posicionamento das unidades de carga na unidade transportadora.

No contexto do PEUC, um bom planejamento possibilita um melhor aproveitamento do espaço disponível, a redução do tempo de embarque e maiores ganhos econômicos para a empresa produtora de celulose.

A Figura 5.11 apresenta algumas fotos das unidades de celulose sendo estivadas em porões de navios. Essas imagens foram obtidas junto a um porto brasileiro especializado no embarque e desembarque de celulose.



FIGURA 5.11 – Estivagem de unidades de celulose em porões de navios

A Figura 5.12(a) apresenta de forma esquemática uma unidade de celulose, a disposição dos porões do navio (b) e uma possível estivagem de um porão(c).

O PEUC, essencialmente, pode ser definido como sendo o problema de empacotar tridimensionalmente itens (unidades) em uma caixa (porão). Uma unidade de celulose é composta por um conjunto de fardos menores, que por sua vez são constituídos de placas de celulose amarradas por um arame. Na literatura, o PEUC pode ser visto como um caso particular de um outro problema mais geral conhecido como *Three-Dimensional Bin Packing Problem* (3D-BPP). No 3D-BPP são dados um conjunto de n itens, cada um caracterizado pela sua largura w_i , altura h_i e comprimento l_i para todo $i \in I = 1, 2, \dots, n$, e um número limitado de *bins* (caixas), no qual inicialmente considera-se uma disponibilidade de pelo menos n bins idênticos e tridimensionais tendo como dimensões comuns a largura W , altura H e comprimento L . O 3D-BPP consiste em se empacotar ortogonalmente todos os n itens no menor número possível de bins (Silva et al., 2003).

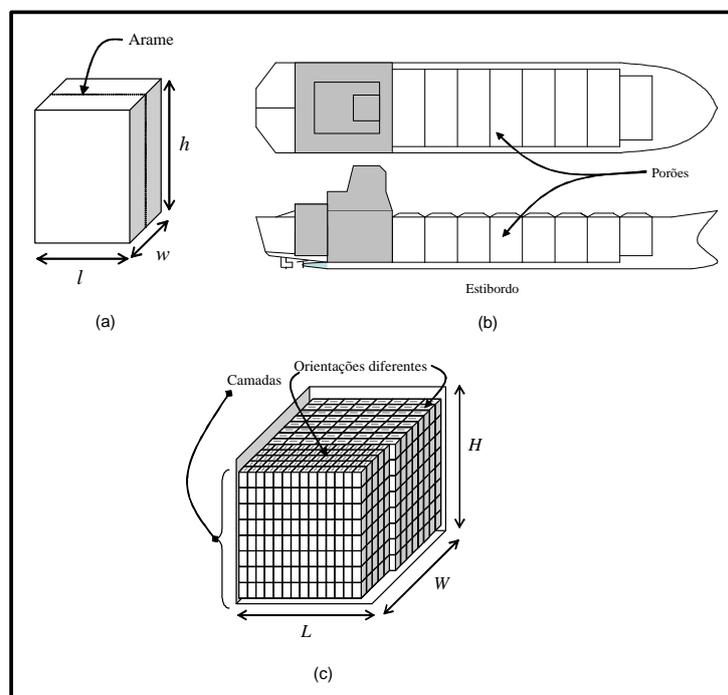


FIGURA 5.12 – Unidade de celulose (a), disposição dos porões e (c) exemplo de uma estivagem

O PEUC é um caso particular do 3D-BPP pois apresenta restrições operacionais que permitem transformá-lo em um problema bidimensional. Todas as unidades de celulose a serem estivadas são iguais, ou seja, apresentam as mesmas dimensões, são do mesmo tipo e devem ser posicionadas em um único porão por vez. Assim, o problema consiste em definir o *layout* da estivagem para cada um dos porões do navio, considerando que as unidades de celulose possuem dimensões iguais inferiores às dimensões do porão.

Em um porão, a estivagem é feita por camadas através dos guindastes, até atingir uma

altura satisfatória que permita fechar a escotilha, e com isso, aproveitar ao máximo o espaço útil do porão.

Cada unidade de celulose também é amarrada por um arame (veja Figura 5.12(a)) para permitir um carregamento mais eficiente no berço de atracação através dos guindastes. Porém, durante o transporte, as unidades tendem a se movimentar devido ao balanço proporcionado pelo navio e, como além das unidades os fardos também são amarrados por arames, se entre uma camada e outra a ordem de colocação das unidades for alterada gerando o cruzamento dos arames, estes podem se partir avariando a carga, produzindo até mesmo incêndios. Assim, para evitar esses problemas, definida a primeira camada do porão, as demais são repetições desta primeira. Com isso o PEUC pode ser reduzido para o caso de empacotamento bidimensional de itens iguais e uma única área maior.

Recorrendo novamente a tipologia proposta por Dyckhoff (1990), o PEUC pode classificado como sendo 2/B/O/C (bidimensional, seleção de itens, objeto único, itens iguais) e com isso, pode ser visto como um PCPP.

Cabe ressaltar que o PEUC apresenta instâncias maiores que as do PCPP, ou seja, enquanto as maiores instâncias do PCPP são da ordem 150 caixas, as do PEUC variam de 150 a 300 unidades de celulose, aumentando assim o seu grau de dificuldade.

5.3.1 Formulação do PEUC

Como o PEUC é classificado como um PCPP, o modelo de Beasley (1985) pode ser utilizado para representá-lo matematicamente. Sendo assim, a formulação 5.6-5.8 pode ser utilizada. Cabe ainda ressaltar que o PEUC possui um número elevado de variáveis e restrições em relação ao PCPP.

5.3.2 LagClus para o PEUC

A aplicação da LagClus no PEUC segue os mesmos passos descritos para o PCPP, inclusive a decomposição das cliques relaxadas e a inserção de subcliques e/ou restrições de adjacência nos *clusters* criados.

Uma vez particionado o grafo de conflitos do PEUC, a formulação 5.9-5.12 pode ser utilizada no PEUC e conseqüentemente as equações 3.8 e 3.9 podem ser novamente consideradas.

Como mencionado no início desta seção, o PEUC apresenta maior dificuldade para ser resolvido do que o PCPP. A Figura 5.13 apresenta um grafo de conflitos para uma

instância real com $(L, W) = (2536\text{cm}, 1312\text{cm})$ e $(l, w) = (144\text{cm}, 84\text{cm})$. São mais de 22 mil vértices e mais de 4,5 milhões de arestas presentes neste grafo.

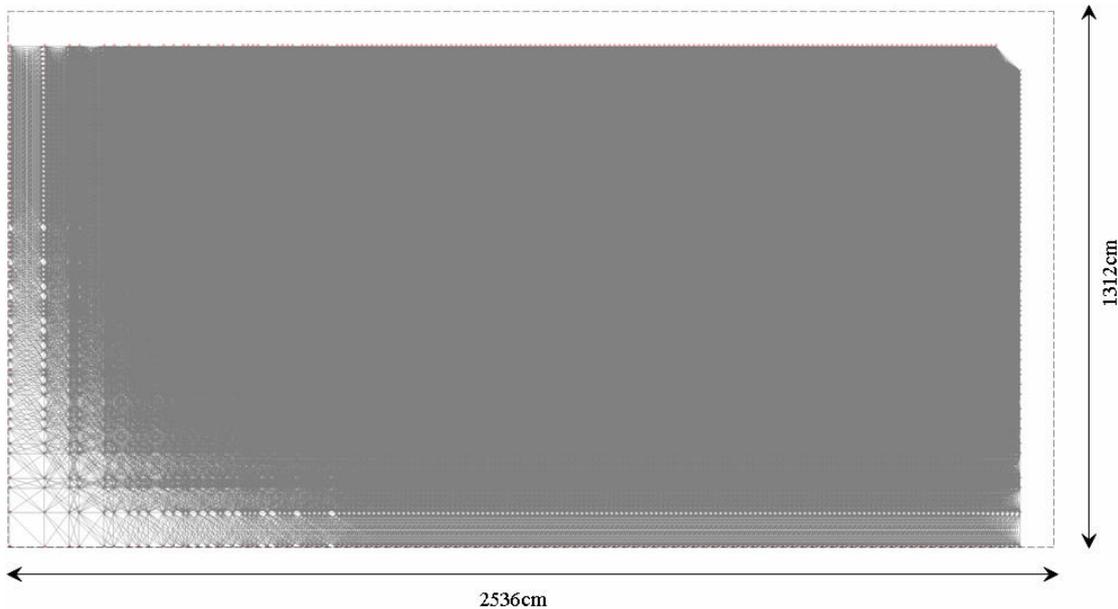


FIGURA 5.13 – Grafo de conflitos gerado para a instância real P2: 22.870 vértices e 4.792.819 arestas

5.3.3 Experimentos Computacionais

O algoritmo de subgradientes descrito para o PCPP foi aplicado novamente no PEUC, com os mesmos critérios de parada. O CPLEX 7.5 foi utilizado para resolver os subproblemas e o METIS para realizar o particionamento do grafo de conflitos.

A heurística lagrangeana HVM foi utilizada para, a partir da solução relaxada da LagClus, compor uma solução viável para o PEUC.

As Tabelas 5.6 e 5.7 mostram, respectivamente, as características dos problemas e os resultados obtidos com a LagClus. Veja que a maioria das soluções práticas ultrapassa 200 unidades de celulose. Essas instâncias foram obtidas em um porto brasileiro especializado no embarque de unidades de celulose.

Na Tabela 5.6, as colunas H e h representam, respectivamente, a altura do porão e altura da unidade de celulose. A última coluna representa o número de unidades estivadas pelo porto atualmente.

Com a formulação do PEUC mostrada na Seção 5.3.1 aplicada diretamente nas instâncias da Tabela 5.6, o CPLEX 7.5 não conseguiu obter resultados inteiros em mais de 10h de

execução. O limitante fornecido pela relaxação de PL também não foi encontrado após o mesmo período de tempo.

TABELA 5.6 – Instâncias consideradas para o PEUC.

Inst.	L (cm)	W (cm)	H (cm)	l (cm)	w (cm)	h (cm)	Solução Prática do Porto
P1	2296	1230	1600	136	94	184	213
P2	2536	1312	1600	144	84	190	269
P3	2252	1470	1652	144	84	190	265
P4	1470	1458	1652	144	84	190	170
P5	2296	1230	1600	135	92	183	217
P6	1804	1230	1600	137	95	190	158
P7	2466	1230	1600	137	95	190	222
P8	1804	1750	1600	137	95	190	234
P9	2426	1230	1640	137	95	190	204

Os resultados da LagClus estão mostrados na Tabela 5.7. Como indicado no Capítulo 3, o número de *clusters* foi previamente definido a partir de experimentos preliminares que avaliaram a qualidade dos limitantes e o tempo computacional.

TABELA 5.7 – Resultados da LagClus para as instâncias do PEUC.

Inst.	# de Clusters	UB _{Area}	UB _{Barnes}	LagClus			
				LB	UB	Gap (%)	Tempo (s)
P1	30	220	220	219	220,01	0,46	4768
P2	20	275	274	273	274,48	0,54	24637
P3	30	273	273	271	272,52	0,56	9634
P4	20	177	177	175	176,28	0,73	2889
P5	30	227	227	226	228,66	1,16	6268
P6	40	170	170	168	170,45	1,44	907
P7	30	233	233	231	232,97	0,85	7460
P8	30	242	242	240	242,77	1,14	16249
P9	30	229	229	227	229,31	1,00	7384

Considere a Tabela 5.7. Para a instância P2 a LagClus forneceu um resultado melhor que o limitante de área. Para as instâncias P3, P4 e P7, a LagClus forneceu limitantes superiores melhores que os já conhecidos de área e de Barnes. Somente na instância P5 o limitante da LagClus não foi bom. Para as demais instâncias os limitantes obtidos (LagClus, de área e de Barnes) podem ser considerados semelhantes. Note que esses resultados podem ser melhorados se o número de *clusters* for reduzido.

Os tempos computacionais da LagClus variaram de 907 a 24637 segundos, porém cabe ressaltar alguns pontos. A LagClus apresenta limitantes que podem ser utilizados para avaliar o quanto longe uma solução viável está da solução ótima. Nos portos, as autoridades portuárias têm armazenado em um banco de dados os planos de estivagem para a sua frota. Então, quando um navio conhecido for atracar no porto, eles imprimem e

utilizam os planos de estivagem previamente armazenados para aquele navio. Além disso, todo navio tem uma programação e eles sabem qual a sua data e horário de chegada. Quando um navio diferente está programado para atracar no porto, eles executam um novo plano de estivagem para aquele navio mediante informações sobre as dimensões dos porões passadas pelo proprietário do navio, conhecido como armador. Normalmente o porto recebe a informação de que um navio diferente vai atracar com meses ou até mesmo 1 ano de antecedência. Portanto, em um ano, as autoridades tem tempo suficiente para executar novos planos de estivagem com métodos mais eficientes como é o caso da LagClus.

TABELA 5.8 – Análise do embarque de unidades de celulose.

Inst.	# de Unidades de Celulose por Camada			# de Unidades de celulose por porão		
	Solução Prática do Porto	LB	Diferença	# de Camadas	Solução Prática do Porto	LB
P1	213	219	6	8	1704	1752
P2	269	273	4	8	2152	2184
P3	265	271	6	8	2120	2168
P4	170	175	5	8	1360	1400
P5	217	226	9	8	1736	1808
P6	158	168	10	8	1264	1344
P7	222	231	9	8	1776	1848
P8	234	240	6	8	1872	1920
P9	204	227	23	8	1632	1816
Média	216,89	225,56	8,67	8	1735,11	1804,44

A Tabela 5.8 faz uma análise geral dos resultados obtidos. A coluna “Diferença” representa a diferença entre a melhor solução dada pela heurística lagrangeana e a solução prática do porto. A coluna “Número de camadas” é calculada como sendo $\lfloor H/h \rfloor$.

Como pode ser observado, a LagClus em média conseguiu estivar 8,67 unidades de celulose a mais por camada, o que dá 69,33 unidades a mais por porão (1804,44 - 1735,11). Considerando uma média de 10 porões por navio, são 693,3 unidades a mais por carregamento por navio. Dado a quantidade de viagens de um navio por ano, o total transportado a mais, pode representar ganhos logísticos significativos em termos de tempo e de custos.

A Figura 5.14 mostra os planos de estivagem usados pelos portos e os aqui obtidos pela LagClus, através da heurística HVM. A parte em cinza indica áreas não usadas (sobras). Note que visualmente as soluções da HVM são melhores que as soluções práticas dos portos.

5.4 Considerações Finais

Este capítulo apresentou a LagClus e o AGC para o Problema do Carregamento de Paletes do Produtor (PCPP). Esse problema pode ser considerado um PMCIIV e consequentemente um grafo de conflitos bem definido é obtido.

A LagClus encontrou soluções ótimas conhecidas de algumas instâncias e forneceu limitantes tão bons quanto os de PL. Por outro lado, o AGC também apresentou bons resultados, encontrando a solução ótima para todas as instâncias consideradas difíceis por Letchford e Amaral (2001).

Está em fase de estudo, a aplicação de técnicas de redução que permitam eliminar colunas improdutivas, e técnicas que parem o processo de geração de colunas, assim que este atingir um determinado valor, como por exemplo, o limitante de área descrito anteriormente. Espera-se com isso, reduzir o tempo computacional e obter problemas mestres restritos finais tão bons quanto os obtidos quando o processo de geração segue até o fim.

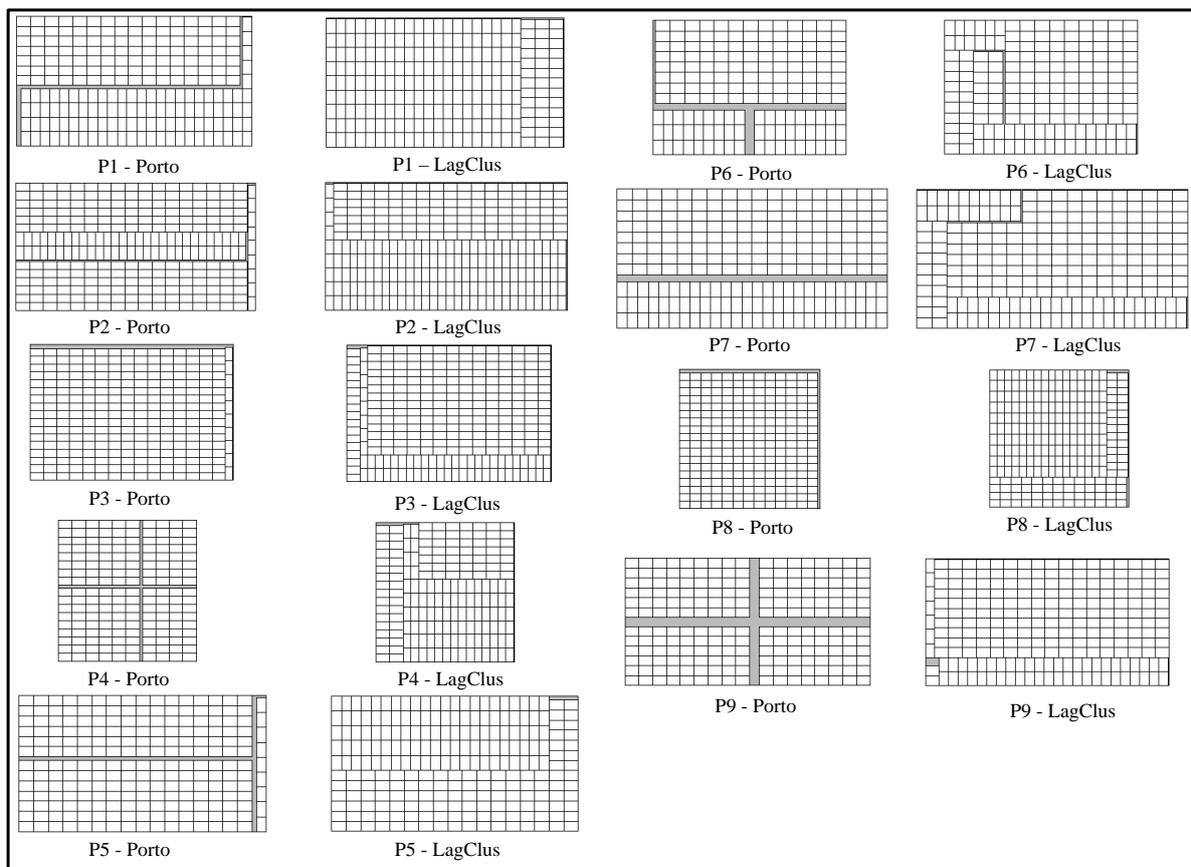


FIGURA 5.14 – Planos de estivagem praticados nos portos e os novos gerados pelo LagClus

Como foi verificado, $v(PPCP_{DW})_{PL}$ encontra o limitante da LagClus mostrando que o método estabiliza.

Neste capítulo também foi apresentado o Problema da Estivagem de Unidades de Celulose (PEUC) em porões de navios dedicados. O PEUC aparece em vários portos brasileiros necessitando de métodos eficientes para resolvê-lo.

Dadas as suas restrições operacionais, esse problema pode ser considerado como um PCPP e assim, a formulação de Beasley (1985) para o problema de corte não-guilhotinado bidimensional pode ser adaptada e utilizada. Como o PEUC apresenta um grau de dificuldade superior ao do PCPP, a LagClus foi testada. Os resultados obtidos foram interessantes e se mostraram superiores aos resultados práticos utilizados atualmente.

O AGC foi testado no PEUC mas os resultados não foram interessantes. A mesma heurística utilizada no PCPP para gerar o conjunto inicial de colunas do PCPP, foi aqui usada. Porém, os resultados computacionais mostraram uma convergência muito lenta, inviabilizando sua aplicação.

Como no PCPP, estratégias eficientes de remoção de colunas precisam ser implementadas para que o AGC possa ser aplicado em problemas maiores como os do PEUC.

O próximo capítulo apresenta uma aplicação da LagClus em um problema no qual o grafo de conflitos não é bem definido.

CAPÍTULO 6

PROGRAMAÇÃO DIÁRIA DE UM SATÉLITE DE OBSERVAÇÃO

Satélites de observação terrestre coletam imagens de áreas específicas do planeta conforme as solicitações de órgãos governamentais, institutos de pesquisas e empresas privadas (Cordeau e Laporte, 2005). Cada imagem tem associado um benefício e satélites típicos não podem acomodar muitas imagens em uma missão devido às restrições operacionais. Assim, se faz necessário selecionar as melhores imagens que maximizem uma função objetivo, respeitando as restrições operacionais.

Existem problemas diferentes relacionados com os satélites de observação terrestre (veja Lin et al. (2005); Benoist e Rottembourg (2004); Cordeau e Laporte (2005); Gabrel e Murat (2003); Habet e Vasquez (2004)). Este capítulo considera o Problema da Programação Diária de Imagens de um Satélite de Observação (PPDISO) correspondente à família dos satélites SPOT. Os satélites SPOT's são satélites ópticos de observação terrestre desenvolvidos pelo CNES (Centro Nacional Francês de Estudos Espaciais). O primeiro satélite (SPOT 1) foi lançado em 1986 e o último (SPOT 5) em 2002 (veja site <http://spot5.cnes.fr>).

O SPOT 5 foi projetado para ser equipado com três câmeras posicionadas na frente, no meio e na parte de trás do satélite. Ele pode obter dois tipos de imagens: *mono* e *stereo*. As imagens *mono* podem ser obtidas por qualquer uma das três câmeras, enquanto uma imagem *stereo* deve ser obtida pelas câmeras da frente e de trás ao mesmo tempo.

No PPDI do satélite SPOT 5 (PPDI-S5) dado um conjunto de imagens candidatas, deve-se selecionar um subconjunto que satisfaz a várias restrições físicas e lógicas (Vasquez e Hao, 2001):

- a. duas tomadas de imagens não devem se sobrepor e o tempo mínimo de transição entre duas sucessivas imagens na mesma câmera deve ser respeitado;
- b. existem limitações quanto ao fluxo de dados instantâneo no satélite que é o resultado de imagens simultâneas obtidas em câmeras diferentes; e
- c. a capacidade de armazenamento do satélite não pode ser excedida.

As instâncias do PPDI-S5 apresentadas na literatura por Bensana et al. (1999), obtidas do simulador do SPOT 5, ainda apresentam-se desafiadoras para *solvers* comerciais.

Sendo assim, este capítulo apresenta algumas relaxações lagrangeanas para o problema, dentre elas a LagClus.

Como o PPDI-S5 não apresenta um grafo de conflitos bem definido, os resultados da LagClus não foram tão interessantes, provando somente a solução ótima de uma instância ainda não conhecida. Por outro lado, ao se analisar a relaxação de PL de um modelo proposto na literatura, identificou-se várias restrições de corte. Com isso, algumas formulações alternativas para o problema foram estudadas. Duas dessas formulações encontraram a solução ótima de todas as instâncias propostas por Bensana et al. (1999).

6.1 Revisão Bibliográfica para o PPDI-S5

Nesta seção são apresentados alguns trabalhos relacionados ao PPDI-S5, bem como outros correlacionados.

Verfaillie et al. (1996) propuseram algoritmos exatos baseados em técnicas de *Branch-and-Bound*. Os autores apresentaram o algoritmo *Russian Doll Search* que realiza várias buscas em subproblemas aninhados, gravando os resultados de cada busca e usando mais tarde para compor um limitante inferior e realizar podas na árvore de busca. Esse algoritmo forneceu bons resultados em problemas reais pequenos. Lemaître e Verfaillie (1997) trabalharam com programação linear inteira com o CPLEX e com programação por restrições com o ILOG Solver.

Bensana et al. (1999) propuseram *benchmarks* para o PPDI-S5 a partir dos dados do simulador do SPOT 5. As instâncias obtidas vem sendo utilizados para comparação na literatura, porém estão divididos em dois conjuntos diferentes: sem restrições de capacidade (Conjunto 1) e com restrições de capacidade (Conjunto 2). O Conjunto 1 é composto de 13 instâncias variando de 67 a 364 imagens candidatas e o Conjunto 2 de 7 instâncias variando de 209 a 1057 imagens candidatas. O algoritmo proposto por Verfaillie et al. (1996) encontrou a solução ótima de todas as instâncias do Conjunto 1, mas falhou nas instâncias no Conjunto 2.

Recentemente, Gabrel (2005, 2006) apresentou resultados computacionais com relaxações de PL de duas formulações binárias para o PPDI-S5: uma proposta por Bensana et al. (1996) e outra por Gabrel e Murat (2003). A formulação proposta por Gabrel e Murat (2003) usa decomposição e análise de fluxo. Conforme Gabrel (2005, 2006), essa formulação fornece limitantes de PL melhores que a formulação de Bensana et al. (1996). Gabrel (2006) ainda descreve técnicas que permitem fortalecer a formulação de Gabrel e Murat (2003).

Por outro lado, muitas heurísticas e metaheurísticas têm sido utilizadas com sucesso no PPDI-S5. Bensana et al. (1996) aplicaram uma Heurística Gulosa, *Simulated Annealing* e Busca Tabu no PPDI-S5, sendo que a Busca Tabu apresentou os melhores resultados.

Vasquez e Hao (2001) propuseram uma formulação binária para o PPDI-S5 que é uma versão generalizada do problema da mochila. Porém, esse modelo apresenta um grande número de restrições binárias e ternárias, dificultando o processo de solução. Os autores também apresentaram uma Busca Tabu mais aprimorada e conseguiram melhorar os resultados de Bensana et al. (1996). Esses resultados atualmente tem sido os melhores até então publicados para as instâncias de Bensana et al. (1999).

Mais tarde, Vasquez e Hao (2003) apresentaram limitantes superiores muito próximos dos limitantes inferiores apresentados pelos mesmos autores em 2001. Esse trabalho permitiu com que os autores provassem que a Busca Tabu anteriormente implementada, realmente fornecera bons resultados.

Para produzir limitantes superiores para o PPDI-S5, Vasquez e Hao (2003) representaram o problema por meio de um grafo que foi particionado em *clusters* e cada um deles resolvido com um algoritmo *Branch-and-Bound* desenvolvido pelos próprios autores. Note que esta idéia é bem semelhante a da LagClus, entretanto, os autores não relaxaram no sentido lagrangeanas as arestas que conectam os *clusters*. Usando o conceito da LagClus, é como se Vasquez e Hao (2003) tivessem utilizado multiplicadores lagrangeanos iguais a zero e executado somente a primeira iteração do algoritmo de subgradientes. Um problema relacionado aos limitantes de Vasquez e Hao (2003) foi o tempo gasto para obtê-los, que variou de algumas horas a alguns dias de processamento.

6.2 Formulação Matemática para o PPDI-S5

Seja $P = p_1, p_2, \dots, p_n$ um conjunto de n imagens candidatas, *mono* ou *stereo*. Cada imagem mono p_i pode ser obtida por qualquer uma das três câmeras, ou seja, existem três combinações possíveis: $(p_i, \text{câmera}_1)$, $(p_i, \text{câmera}_2)$ e $(p_i, \text{câmera}_3)$. Similarmente, cada imagem *stereo* p_i pode ser obtida somente pelas câmeras da frente e de trás ao mesmo tempo, ou seja, existe apenas uma combinação possível $(p_i, \text{câmera}_{1,3})$. Se n_1 representa o total de imagens *mono* e n_2 o total de imagens *stereo*, existe um total de $m = 3 \times n_1 + n_2$ pares possíveis de imagens e câmeras. Cada possível par pode ser representado por uma variável binária x_i para todo $i = 1, 2, \dots, m$.

Cada imagem p_i tem associado um benefício b_i e um peso c_i . Para as imagens *mono* p_i , cada uma das variáveis correspondentes aos seus pares $(p_i, \text{câmera})$, recebem o mesmo

benefício b_i e peso c_i . Assim, o modelo proposto por Vasquez e Hao (2001) pode ser assim descrito:

$$v(PPDI - S5) = Max \left\{ \sum_{i=1}^m b_i x_i \right\} \quad (6.1)$$

(PPDI - S5) Sujeito a:

$$\sum_{i=1}^m c_i x_i \leq Cap \quad (6.2)$$

$$x_i + x_j \leq 1 \quad \forall (i, j) \in C_1 \quad (6.3)$$

$$x_i + x_j + x_k \leq 1 \quad \forall (i, j, k) \in C_2 \quad (6.4)$$

$$x_i + x_j + x_k \leq 2 \quad \forall (i, j, k) \in C_3 \quad (6.5)$$

$$x_i \in \{0, 1\} \quad \forall i = 1 \dots m \quad (6.6)$$

Sendo:

- Cap a capacidade máxima de armazenamento *on board*;
- C_1 um conjunto de variáveis x_i e x_j tal que, devido à incompatibilidades entre as imagens i e j , estas não podem ser tomadas;
- C_2 um conjunto de três variáveis (x_i, x_j e x_k) em que no máximo uma delas pode ser selecionada; e
- C_3 um conjunto de três variáveis (x_i, x_j e x_k) em que no máximo duas delas podem ser selecionadas.

A restrição 6.2 indica que a soma dos pesos das imagens selecionadas (programadas para serem obtidas pelo satélite) não pode exceder a capacidade máxima Cap de armazenamento do satélite. Essa restrição é também conhecida como restrição de mochila.

As restrições definidas em 6.3 são lógicas e garantem:

- a. a não sobreposição de imagens na mesma câmera; e
- b. limitações de fluxo instantâneo no satélite.

Note que as restrições definidas em 6.3 são restrições de adjacência ou restrições binárias.

Algumas restrições envolvendo limitações de fluxo instantâneo não podem ser expressas pelas restrições binárias 6.3, então um conjunto complementar de restrições deve ser definido. Esse novo conjunto está definido em 6.5 e suas restrições são conhecidas como restrições ternárias.

As restrições definidas em 6.4 garantem que uma imagem *mono* será tomada por no máximo uma câmera e as restrições 6.6 indicam que todas as variáveis são binárias. Veja que as restrições 6.4 são cliques com três variáveis e que também podem ser expressas por três restrições binárias como em 6.3.

Como indicado por Vasquez e Hao (2001), a formulação acima é um caso especial do problema da mochila multidimensional (Martello e Toth, 1990) e tem recebido atenção por parte dos pesquisadores devido à suas aplicações práticas (veja Fréville e Hanafi (2005)).

Embora o modelo 6.1-6.6 seja eficiente em instâncias de pequeno porte sem a restrição de capacidade, ele apresenta problemas com as instâncias maiores que possuem restrição de mochila.

Esse modelo apresenta um limitante de PL ruim (veja Vasquez e Hao (2003)) não permitindo que a técnica de *Branch-and-Bound* realize as podas mais rapidamente para garantir a solução ótima de uma instância. Sendo assim, algumas relaxações lagrangeanas são apresentadas na próxima seção buscando identificar novas soluções ótimas.

6.3 Relaxações Lagrangeanas para o PPDI-S5

Como visto no Capítulo 2, o primeiro passo antes de aplicar uma relaxação Lagrangeana é identificar o conjunto de restrições que dificultam o processo de solução.

Experimentos preliminares mostraram que as restrições ternárias são as que apresentam maior complexidade, ou seja, quando estas restrições são removidas do modelo, *solvers* comerciais conseguem resolver o problema relaxado, mesmo com a restrição de capacidade. Sendo assim, além da LagClus, optou-se também por realizar uma relaxação lagrangeana sobre as restrições 6.5.

6.3.1 LagClus para o PPDIS-S5

Analisando o modelo 6.1-6.6 percebe-se que restrições de adjacência aparecem em 6.3 e 6.4. Seja $G = (V, A)$ um grafo de conflitos no qual o conjunto de vértices V é formado

pelas variáveis binárias do modelo de Vasquez e Hao (2001), e o conjunto de arestas A pelas incompatibilidades descritas pelas restrições de adjacência definidas em 6.3 e 6.4.

O grafo G acima não representa o PPDI-S5, pois a restrição de capacidade nem as restrições ternárias foram consideradas. Por outro lado, como indicou Vasquez e Hao (2003), o número de restrições binárias é muito maior que o número de restrições ternárias, conseqüentemente, adotou-se trabalhar as restrições ternárias como “restrições de adjacência” para “considerá-las” em G .

Assim, a LagClus para o PPDI-S5 pode ser assim definida:

- Relaxe a restrição de capacidade no sentido lagrangeano;
- Crie um grafo de conflitos $G = (V, A)$ no qual cada variável binária do modelo 6.1-6.6 é representado por um vértice em G e as restrições definidas em 6.3, 6.4 e 6.5 são representadas pelas arestas em G ;
- Aplique uma heurística de particionamento para dividir G em \bar{P} partes, formando \bar{P} clusters;
- Relaxe no sentido lagrangeano todas as arestas que conectam os clusters referentes às restrições binárias e todas as restrições ternárias que apresentam vértices em diferentes clusters; e
- A relaxação lagrangeana é decomposta em \bar{P} subproblemas e resolvida.

A Figura 6.1 apresenta os passos acima descritos. Note que o grafo da figura apresenta restrições binárias representadas por linhas pontilhadas, e restrições ternárias por linhas contínuas. Esse grafo foi particionado em dois clusters.

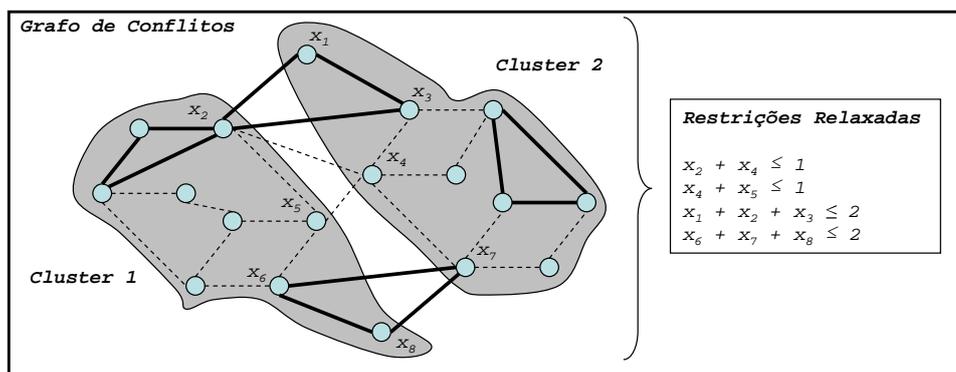


FIGURA 6.1 – Grafo de conflitos com restrições binárias e ternárias.

No exemplo da Figura 6.1, repare que duas restrições ternárias foram relaxadas porque um de seus vértices aparece em um *cluster* diferente dos demais. Também repare que duas arestas de adjacência são relaxadas. Elas poderiam ter sido obtidas da seguinte clique $x_2 + x_4 + x_5 \leq 1$, e neste caso a aresta (x_2, x_5) não é relaxada e a respectiva restrição de adjacência $x_2 + x_5 \leq 1$ deve ser adicionada no *Cluster* 1.

Ao dividir o grafo de conflitos conforme os passos acima, o modelo 6.1-6.6 pode ser assim escrito:

$$v(PPDI - S5) = \text{Max} \sum_{p=1}^{\bar{P}} b^p x^p \quad (6.7)$$

(PPDI-S5) Sujeito a:

$$+A^1 x^1 \quad +A^2 x^2 \quad \dots \quad +A^{\bar{P}} x^{\bar{P}} \leq R^A \quad (6.8)$$

$$\begin{aligned} +D^1 x^1 & \leq R^1 \\ & +D^2 x^2 & \leq R^2 \\ & \dots & \leq \vdots \\ & & +D^{\bar{P}} x^{\bar{P}} & \leq R^{\bar{P}} \end{aligned} \quad (6.9)$$

$$x^1 \in B^{n_1} \quad \dots \quad \dots \quad x^{\bar{P}} \in B^{n_{\bar{P}}} \quad (6.10)$$

Sendo:

- A^p uma matriz que representa os coeficientes associados às variáveis x^p do *cluster* p que aparecem nas restrições: de capacidade, de adjacência 6.3 com vértices entre *clusters*, de cliques 6.4 com vértices entre *clusters* e ternárias 6.5 com vértices em *clusters* diferentes. O número de linhas dessa matriz depende do número de restrições relaxadas;
- D^p uma matriz binária que representa os coeficientes associados às variáveis x^p que aparecem nas restrições definidas em 6.3-6.5 do *cluster* p ;
- B^{n_p} um vetor de variáveis binárias com dimensão n_p associadas às variáveis x^p ;

- R_A um vetor com os coeficientes do lado direito das restrições relaxadas; e
- R_p um vetor com os coeficientes do lado direito das restrições definidas em 6.3-6.5 pertencentes ao *cluster* p .

Agora usando multiplicadores lagrangeanos $\lambda \in \mathfrak{R}_+^M$, sendo M o total de restrições relaxadas 6.8, a LagClus pode ser assim escrita:

$$v(LG_\lambda PPDI - S5) = \sum_{p=1}^{\bar{P}} v(PPDI - S5)_p + \sum_{m=1}^M R_m^A \lambda_m \quad (6.11)$$

Onde cada subproblema p pode ser assim definido:

$$v(PPDI - S5)_p = Max \{ (1 - A^{Tp} \lambda) x^p : x^p \in D_p \} \quad (6.12)$$

6.3.2 Relaxação Lagrangeana Sobre as Restrições Ternárias

Seja $C = |C_3|$ o total de restrições ternárias definidas em 6.5. Utilizando multiplicadores $\lambda \in \mathfrak{R}_+^C$, as restrições ternárias são relaxadas no sentido lagrangeano produzindo o seguinte modelo:

$$v(L_\lambda PPDI - S5) = Max \left\{ \sum_{i=1}^m b_i x_i + \sum_{\substack{c=1 \\ \{(i,j,k) \in C_{3,c}\}}}^C \lambda_c (-x_i - x_j - x_k) + 2 \sum_{c=1}^C \lambda_c \right\} \quad (6.13)$$

($L_\lambda PPDI-S5$) Sujeito a: 6.2-6.4 e 6.6.

Os resultados das relaxações lagrangeanas apresentadas para o PPDI-S5, de maneira geral, não foram bons. Apesar da LagClus provar uma solução ótima ainda não conhecida, os *gaps* encontrados para as instâncias com restrição de capacidade piores que os de Vasquez e Hao (2003). Porém os estudos aqui desenvolvidos incentivaram a criação de modelos matemáticos mais fortes para o PPDI-S5 que utilizassem restrições de corte. Esses modelos são mostrados a seguir.

6.4 Novas Formulações para o PPDI-S5

Todas as formulações aqui propostas foram obtidas a partir da relaxação de PL do modelo Vasquez e Hao (2001). Como descrito no Capítulo 3, existem vários cortes já estudados na literatura, porém dependendo do problema, novos cortes podem ser encontrados e

implementados.

6.4.1 Formulação Usando Cliques Maximais Envolvendo um Vértice e Cobertura de Vértices

Murray e Church (1997a) trabalhando com o *PMClV* propuseram uma nova forma de representar restrições de adjacências, que permite reduzir o número de restrições de um modelo. Eles exploram o conceito de cliques maximais e de cobertura de vértices. Uma clique maximal é definida por Murray e Church (1997a) como sendo a maior clique que envolve um determinado vértice. Por exemplo, considere a Figura 6.2 que apresenta um grafo de conflitos. A maior clique que envolve o vértice x_1 é dado pelo conjunto de vértices x_1, x_2, x_3 e x_4 .

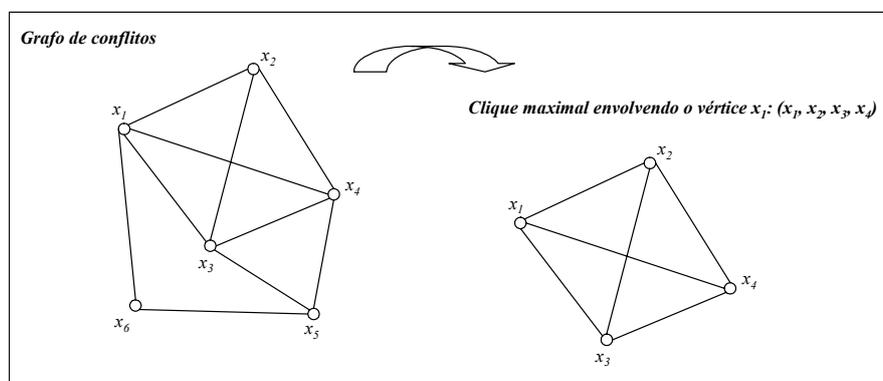


FIGURA 6.2 – Clique maximal envolvendo o vértice x_1 .

Entretanto, um conjunto de restrições baseadas em cliques maximais podem conter restrições (cliques) redundantes. Isso significa que uma clique maximal de um vértice pode ser um subconjunto de uma clique maximal de outro vértice. Com isso, faz-se necessário identificar somente cliques maximais únicas para evitar redundância. Esse processo de identificação deve ser feito comparando-se as novas cliques maximais obtidas com as que foram identificadas previamente. Com isso, pode-se dizer que o número de cliques maximais únicas é limitado pelo número de vértices presentes no grafo de conflitos (Murray e Church, 1997a).

Porém, ao selecionar o conjunto de cliques maximais, não existe garantia de que todas as restrições de adjacências estejam satisfeitas. Considerando novamente a Figura 6.2, a aresta (x_1, x_6) não faz parte da clique maximal do vértice x_1 , mostrada anteriormente. Com isso, faz-se necessário adicionar às restrições de cliques maximais, restrições de cobertura de vértices ou de empacotamento de vértices. Assim, são adicionadas, no máximo, um número de restrições de cobertura igual ao número de vértices do grafo

de conflitos (Murray e Church, 1997a).

A Figura 6.3 apresenta a redução proposta por Murray e Church (1997a) aplicada ao grafo mostrado na Figura 6.2. Observe que o número de restrições obtido ao final é igual a 4, dividido em 2 restrições de cliques maximais envolvendo os vértices x_1 e x_5 , e 2 restrições de cobertura envolvendo os vértices x_1 e x_5 . Observe ainda que este número de restrições é inferior a 2 vezes o números de vértices conforme Murray e Church (1997a).

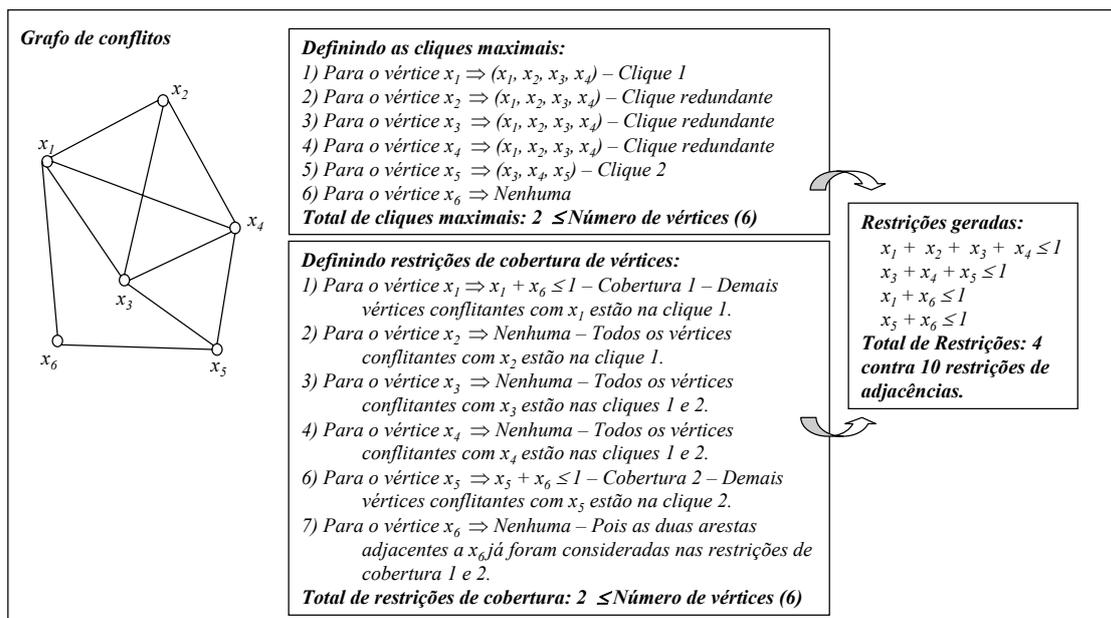


FIGURA 6.3 – Exemplo de aplicação da redução de Murray e Church (1997a).

6.4.1.1 Definição das Restrições de Redução

Seja CL_k o conjunto de vértices pertencentes à clique maximal k , como definido por Murray e Church (1997a), para $k = \{1, \dots, K\}$, tal que K representa o número total de cliques maximais. Assim, as restrições de cliques maximais podem ser expressas como:

$$\sum_{j \in CL_k} x_j \leq 1 \quad \forall k = \{1, \dots, K\} \quad (6.14)$$

Seja N_i um conjunto com todos os vértices que compartilham uma aresta com o vértice i , e n_i um coeficiente igual ao maior número de elementos do conjunto N_i que podem ser selecionados simultaneamente sem violar uma restrição de adjacência. Sendo assim, as restrições de cobertura de vértices podem ser definidas como:

$$n_i x_i + \sum_{j \in N_i} x_j \leq n_i \quad \forall i = \{1, \dots, V\} \quad (6.15)$$

Sendo V o conjunto de vértices do grafo de conflitos formado pelas restrições de adjacência presentes no modelo de Vasquez e Hao (2001).

Porém dessa forma, as restrições definidas em 6.14 aparecem também nas restrições de vértices definidas em 6.15, indicando redundância de restrições. Para evitar isso, seja $\hat{N}_i = \{j \in N_i \mid j \notin CL_k \forall k = 1, \dots, K\}$ o conjunto dos vértices j que compartilham uma aresta com i , dado que a aresta (i, j) não aparece em nenhuma clique maximal k , e \hat{n}_i um coeficiente igual ao maior número de elementos do conjunto \hat{N}_i que podem ser selecionados simultaneamente. Sendo assim, a restrição definida em 6.15 pode ser substituída por:

$$\hat{n}_i x_i + \sum_{j \in \hat{N}_i} x_j \leq \hat{n}_i \quad \forall i = \{1, \dots, V\} : \hat{N}_i \neq \emptyset \quad (6.16)$$

Com isso, as restrições definidas em 6.14 e em 6.16 podem substituir, sem perda de generalidade, o conjunto de restrições de adjacências presentes em um grafo de conflitos.

6.4.1.2 Algoritmo de Redução

O processo de geração das restrições definidas em 6.14 e 6.16 está apresentado na Figura 6.4.

Este processo é constituído de duas fases, diferenciadas pelos retângulos maiores. O primeiro passo é identificar as cliques maximais. Isso é feito para cada vértice i e consiste em uma procura no conjunto N_i do maior subconjunto de vértices (incluindo i) que é mutuamente conectado. O segundo passo consiste no processo de criação das restrições de cobertura de vértices que constitui das restrições de adjacências não consideradas no passo anterior, e na definição do coeficiente \hat{n}_i .

Ao término do algoritmo mostrado na Figura 6.4, as restrições obtidas podem substituir todas as restrições de adjacências presentes em um grafo de conflitos.

Na próxima seção, será apresentada uma formulação reduzida para o PPDI que utiliza os conceitos apresentados nesta seção.

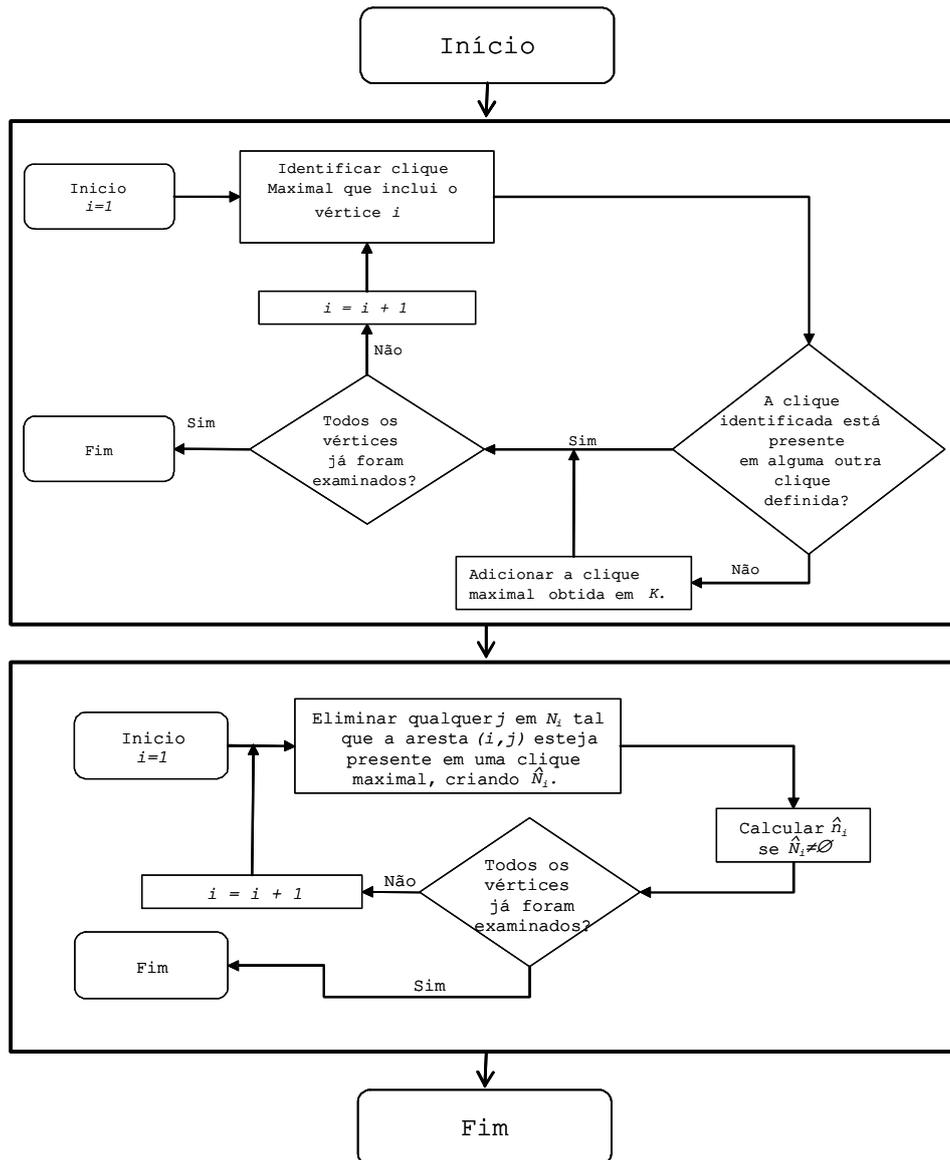


FIGURA 6.4 – Processo de geração das restrições de Murray e Church (1997a).

6.4.1.3 Formulação Usando Uma Clique Maximal para Cada Vértice e Cobertura de Vértices (PPDI-S5_M1)

As restrições definidas em 6.3 e 6.4, podem ser reduzidas através da formulação de Murray e Church (1997a). Com isso, o problema original do PPDI-S5 pode ser reescrito como:

$$v(PPDI - S5_M_1) = Max \left\{ \sum_{i=1}^m b_i x_i \right\} \quad (6.17)$$

(PPDI - S5_M1) Sujeito a:

$$\sum_{i=1}^m c_i x_i \leq Cap \quad (6.18)$$

$$\sum_{j \in CL_k} x_j \leq 1 \quad \forall k \in K \quad (6.19)$$

$$\hat{n}_i x_i + \sum_{j \in \hat{N}_i} x_j \leq \hat{n}_i \quad \forall i = \{1, \dots, V\} : \hat{N}_i \neq \emptyset \quad (6.20)$$

$$x_i + x_j + x_k \leq 2 \quad \forall (x_i, x_j, k_k) \in C_3 \quad (6.21)$$

$$x_i \in \{0, 1\} \quad \forall i \in \{1, \dots, m\} \quad (6.22)$$

Este modelo, apesar de reduzir o número de restrições, não apresenta bons resultados. O CPLEX apresenta melhores resultados com o modelo-base de Vasquez e Hao (2001).

6.4.2 Formulação Usando Cliques Maximais (PPDI-S5_M2)

Considerando o grafo de conflitos $G = (V, A)$ definido com as restrições 6.3 e 6.4, cliques maximais em G podem ser obtidas, fortalecendo o modelo (Nemhauser e Sigismondi, 1992). Uma clique k é dita ser maximal quando nenhuma outra clique a domina.

Enquanto Murray e Church (1997a) consideravam apenas uma clique maximal por vértice, aqui todas as cliques maximais presentes em G devem ser obtidas.

Assim, o segundo modelo, denominado PPDI-S5_M2, pode ser apresentado:

$$v(PPDI - S5_M2) = Max \left\{ \sum_{i=1}^m b_i x_i \right\} \quad (6.23)$$

(PPDIS-S5_M2) Sujeito a:

$$\sum_{i=1}^m c_i x_i \leq Cap \quad (6.24)$$

$$\sum_{i \in k} x_i \leq 1 \quad \forall \text{Clique Maximal } k = 1, \dots, K \quad (6.25)$$

$$x_i + x_j + x_k \leq 2 \quad \forall (i, j, k) \in C_3 \quad (6.26)$$

$$x_i \in \{0, 1\} \quad \forall i = 1 \dots m \quad (6.27)$$

Onde K é o total de cliques maximais encontradas no grafo G .

A função objetivo do modelo PPDI-S5_M2 é a mesma de Vasquez e Hao (2001). As restrições 6.24 e 6.26 são as mesmas descritas em 6.2 e 6.5, respectivamente. Fulkerson (1971) e Padberg (1973) indicam que o conjunto de restrições 6.25 é forte, e que o limitante de PL é consequentemente melhorado.

A formulação PPDI-S5_M2 apresenta limitantes de PL melhores que os da formulação de Vasquez e Hao (2001).

6.4.3 Formulação Usando Cliques Maximais e Situação Par-Binário (PPDI-S5_M3)

O modelo PPDI-S5_M2, embora apresente limitantes de PL melhores, ainda consideram as restrições ternárias de maneira independente. Analisando a relaxação de PL do modelo PPDI-S5_M2, essas restrições podem ser fortalecidas da seguinte maneira.

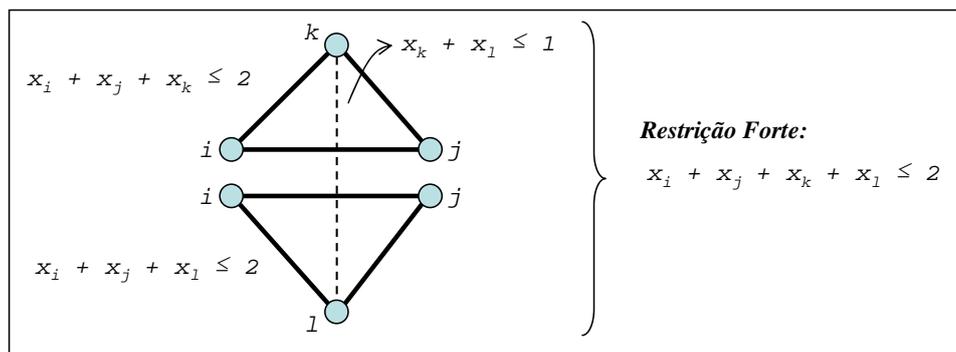


FIGURA 6.5 – Desigualdade válida para o PPDI-S5.

Considere o exemplo mostrado na Figura 6.5. Os dois triângulos são representações de duas restrições ternárias, e a linha pontilhada uma restrição de adjacência definida por 6.25. Essa tipo de situação aqui é denominada par-binário. A solução fracionada $[x_i \ x_j \ x_k \ x_l] = [0,7 \ 0,8 \ 0,5 \ 0,5]$ satisfaz todas as três restrições. Usando o procedimento de Chvátal-Gomory mostrado no Capítulo 2, uma inequação válida pode ser obtida de tal maneira que esta solução fracionada seja eliminada. Assim, se as

três restrições da figura forem somadas obtém-se a seguinte desigualdade $2x_i + 2x_j + 2x_k + 2x_l \leq 5$. Se agora esta inequação for dividida por 2, surge uma nova restrição $x_i + x_j + x_k + x_l \leq 5/2$, entretanto, o lado esquerdo dessa restrição nunca será maior que 2, então a seguinte restrição mais forte é válida:

$$x_i + x_j + x_k + x_l \leq 2. \quad (6.28)$$

A restrição 6.28 domina as demais restrições ternárias presentes na Figura 6.5 que podem ser eliminadas agora.

Seja Q um conjunto com todas as restrições de par-binário. Assim, uma terceira formulação denominada PPDIS-S5_M3 pode ser apresentada:

$$v(PPDI - S5_M3) = Max \left\{ \sum_{i=1}^m b_i x_i \right\} \quad (6.29)$$

(PPDIS-S5_M3) Sujeito a:

$$\sum_{i=1}^m c_i x_i \leq Cap \quad (6.30)$$

$$\sum_{i \in k} x_i \leq 1 \quad \forall \text{Clique Maximal } k = 1, \dots, K \quad (6.31)$$

$$\sum_{i \in q} x_i \leq 2 \quad \forall \text{Par - Binário } q = 1, \dots, Q \quad (6.32)$$

$$x_i + x_j + x_k \leq 2 \quad \forall (i, j, k) \in \overline{C_3} \quad (6.33)$$

$$x_i \in \{0, 1\} \quad \forall i = 1 \dots m \quad (6.34)$$

Sendo $\overline{C_3}$ um conjunto de restrições ternárias não dominadas pelas restrições de par-binário definidas em 6.32.

6.4.4 Formulação Usando Cliques Maximais e Ternárias com Cliques (PPDI-S5_M4)

Os dois últimos modelos fornecem bons limitantes de PL, mas ainda não melhoram o processo de otimização quando se utiliza um *solver* comercial, motivando o estudo de uma quarta formulação.

As restrições 6.32 podem ser generalizadas da seguinte maneira. Suponha que existe um conjunto de restrições ternárias tal que:

- a. todas elas possuam duas variáveis x_i e x_j :

$$x_i + x_j + x_k \leq 2 \quad \forall k \in K ; e$$

- b. o conjunto de variáveis restantes x_k pertencem a uma clique maximal definida em 6.31: $\sum_{k \in K} x_k \leq 1$.

Sob essas condições, a seguinte inequação é válida: $x_i + x_j + \sum_{k \in K} x_k \leq 2$.

Para provar isso, note que $\sum_{k \in K} x_k \leq 1$ pode receber somente os valores 0 e 1. No primeiro caso, a inequação se reduz para $x_i + x_j \leq 2$, o que é obviamente válido. No segundo caso, existe uma variável $x_k = 1$ com $k \in K$. Consequentemente, a restrição $x_i + x_j + x_k \leq 2$ força $x_i + x_j \leq 1$, então $x_i + x_j + \sum_{k \in K} x_k \leq 2$ é válida. Essa restrição é denominada aqui de ternária com cliques.

Considere o exemplo mostrado na Figura 6.6. Os triângulos com linhas escuras representam quatro restrições ternárias, e as linhas pontilhadas, seis restrições de adjacência que podem ser definidas por 6.3, 6.4 ou por uma clique maximal 6.31. Agora a partir das restrições binárias, pode-se deduzir a clique $x_k + x_l + x_m + x_n \leq 1$, e assim pode-se deduzir a seguinte inequação válida: $x_i + x_j + x_k + x_l + x_m + x_n \leq 2$.

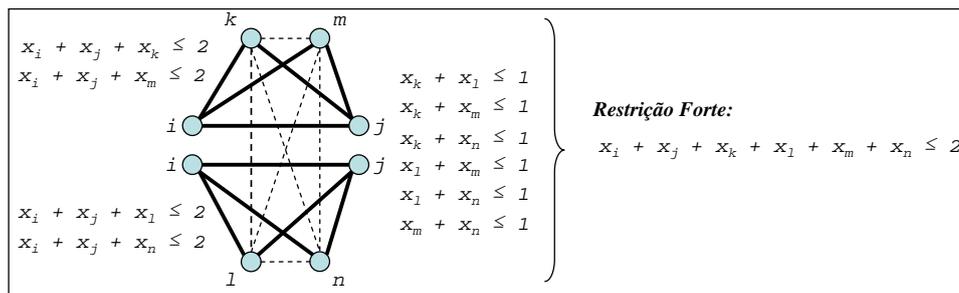


FIGURA 6.6 – Inequações válidas para o PPDI-S5 usando restrições ternárias e cliques.

Essa restrição pode ser também deduzida a partir do procedimento de Chvátal-Gomory. Ao somar as restrições ternárias e adicionar 3 vezes a restrição de clique, obtém-se a seguinte restrição $4x_i + 4x_j + 4x_k + 4x_l + 4x_m + 4x_n \leq 11$. Agora se esta restrição for dividida por 4, a seguinte restrição é obtida $x_i + x_j + x_k + x_l + x_m + x_n \leq 11/4$. Usando a mesma idéia apresentada anteriormente, encontra-se a restrição definida acima.

Seja T um conjunto com as restrições ternárias com cliques como descrito na Figura 6.6. Para encontrar este conjunto, selecione para cada par (x_i, x_j) de uma restrição ternária, todas as variáveis que separadamente compõem uma restrição ternária com x_i e x_j , em seguida, encontre todas as cliques maximais com esses vértices. Cada clique maximal encontrada deve ser inserida em T juntamente com x_i e x_j .

Agora uma quarta formulação para o PPDI-S5, denominada PPDI-S5_M4, pode ser apresentada:

$$v(PPDI - S5_M4) = Max \left\{ \sum_{i=1}^m b_i x_i \right\} \quad (6.35)$$

(PPDIS-S5_M4) Sujeito a:

$$\sum_{i=1}^m c_i x_i \leq Cap \quad (6.36)$$

$$\sum_{i \in k} x_i \leq 1 \quad \forall \text{Clique Maximal } k = 1, \dots, K \quad (6.37)$$

$$\sum_{i \in t} x_i \leq 2 \quad \forall \text{Ternária com Cliques } t = 1, \dots, T \quad (6.38)$$

$$x_i + x_j + x_k \leq 2 \quad \forall (i, j, k) \in \overline{C_3} \quad (6.39)$$

$$x_i \in \{0, 1\} \quad \forall i = 1 \dots m \quad (6.40)$$

Sendo $\overline{C_3}$ um conjunto de restrições ternárias não dominadas pelas restrições de ternária com cliques definidas em 6.38.

6.4.5 Formulação Usando Cliques Maximais e “Cliques Maximais Ternárias” a partir de Restrições Binárias (PPDI-S5_M5)

Considere novamente a Figura 6.6 e note a seguinte característica presente no exemplo. Se todas as restrições ternárias forem selecionadas a partir de cada restrição binária, outra inequação forte pode ser obtida denominada aqui de “clique ternária”.

Considere agora um conjunto de variáveis $x_k, k \in K$ tal que existe uma restrição ternária ligando cada três variáveis nesse conjunto, isto é, $x_i + x_j + x_l \leq 2$ para todas as triplas de distintos índices i, j e $l \in K$. O conjunto K (bem como o conjunto de variáveis indexados por K) é aqui denominado de “clique ternária”. As restrições de “cliques ternárias” $\sum_{k \in K} x_k \leq 2$ são válidas para o PPDI-S5.

Para provar isso, suponha que $\sum_{k \in K} x_k > 2$. Então, uma vez que as variáveis são 0-1, deve existir pelo três variáveis $x_i = x_j = x_l = 1$, com i, j e $l \in K$. Essa situação é impossível uma vez que a restrição ternária $x_i + x_j + x_l \leq 2$ seria violada. Sendo assim, conclui-se que $\sum_{k \in K} x_k \leq 2$ deve ser satisfeita.

Observe que, dentre as restrições de “cliques ternárias”, aquelas que não são dominadas são definidas como “cliques maximais ternárias” K .

Sendo assim, para gerar mais “cliques maximais ternárias”, o conjunto de restrições ternárias 6.5 deve ser aumentado da seguinte maneira. A partir de cada restrição binária $x_i + x_j \leq 1$, um conjunto de restrições ternárias $x_i + x_j + x_l \leq 2$ pode ser obtido somente adicionando $x_l \leq 1$ para qualquer l . Dentre todas as restrições ternárias geradas, as interessantes são as obtidas a partir da restrição binária $x_i + x_j \leq 1$ tal que ao menos uma das variáveis x_i ou x_j esteja presente nas restrições ternárias 6.5.

Considere a Figura 6.7 na qual existem 4 restrições ternárias e 5 restrições binárias. Se a partir da restrição binária $x_i + x_j \leq 1$, todas as restrições ternárias com x_i ou x_j forem selecionadas, novas restrições ternárias podem ser criadas como mostrado na figura. Assim, as 4 restrições ternárias podem ser substituídas por uma única válida e mais forte restrição de clique ternária:

$$x_i + x_j + x_k + x_l + x_m + x_n \leq 2 \quad (6.41)$$

Por exemplo, a seguinte solução fracionada:

$$[x_i \ x_j \ x_k \ x_l \ x_m \ x_n] = [0,5 \ 0,5 \ 0,75 \ 0,75 \ 0,25 \ 0,25]$$

satisfaz todas as restrições presentes na Figura 6.7. Após substituí-las pela clique ternária definida em 6.41, essa solução não é mais permitida.

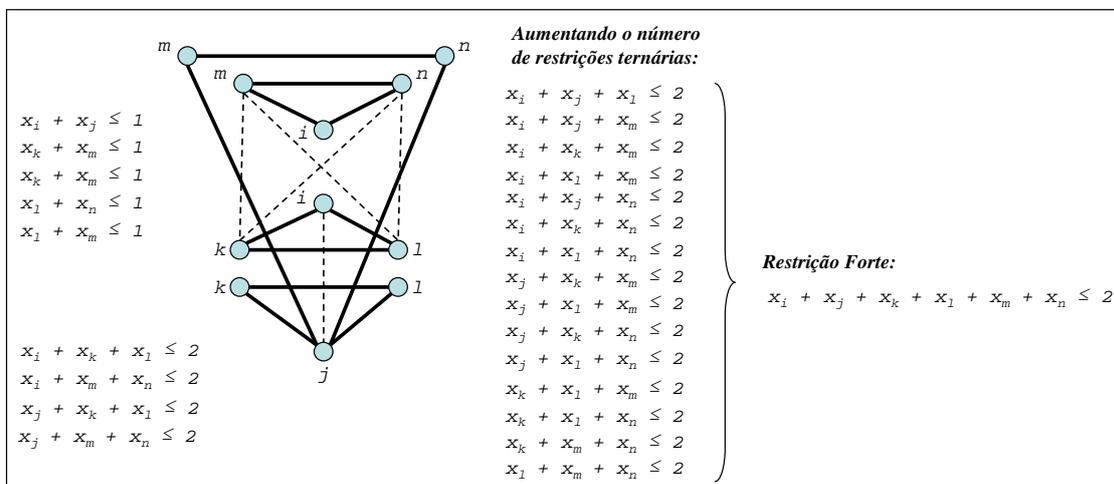


FIGURA 6.7 – Inequação válida para o PPDI-S5 usando as “cliques maximais ternárias” a partir das restrições binárias.

Seja Ψ um conjunto com as restrições de “cliques maximais ternárias”. Para encontrar este conjunto, pode-se aplicar o algoritmo descrito na figura abaixo.

```

Algoritmo Gera_Cliq_Max_Ter_de_Rest_Binárias()
// Seja:
// -  $\Psi$  um conjunto de “cliques maximais ternárias”
// - S um conjunto de variáveis

1   $\Psi \leftarrow \emptyset$ 
2  Para cada restrição  $(x_i, x_j) \in C_1$  ou  $C_2$  Faça
3     $S \leftarrow \emptyset$ ;
4    Selecione todas as restrições ternárias com  $x_i$  ou  $x_j$  e
    adicione suas variáveis em S
5    Com as variáveis em S, gere todas as “cliques ternárias” e
    as adicione em  $\Psi$ 
6  Fim Para
7  Remova todas as “cliques ternárias” em  $\Psi$  que estão dominadas
    
```

FIGURA 6.8 – Algoritmo para gerar as “cliques maximais ternárias” a partir de restrições binárias.

Embora o conjunto de “cliques maximais ternárias” definido com o algoritmo acima domina todas as restrições par-binário, ele não domina C_3 . Sendo assim, uma nova formulação, denominada PPDI-S5_M5, pode ser apresentada:

$$v(PPDI - S5_M_5) = Max \left\{ \sum_{i=1}^m b_i x_i \right\} \quad (6.42)$$

(PPDIS-S5_M5) Sujeito a:

$$\sum_{i=1}^m c_i x_i \leq Cap \quad (6.43)$$

$$\sum_{i \in k} x_i \leq 1 \quad \forall \text{Clique Maximal } k = 1, \dots, K \quad (6.44)$$

$$\sum_{i \in \psi} x_i \leq 2 \quad \forall \text{Clique Maximal Ternária} \quad (6.45)$$

(obtida a partir de restrições binárias) $\psi = 1, \dots, \Psi$

$$x_i + x_j + x_k \leq 2 \quad \forall (i, j, k) \in \overline{C_3} \quad (6.46)$$

$$x_i \in \{0, 1\} \quad \forall i = 1 \dots m \quad (6.47)$$

Sendo $\overline{C_3}$ um conjunto de restrições ternárias não dominadas pelas restrições de ternária com cliques definidas em 6.45.

6.4.6 Formulação Usando Cliques Maximais e “Cliques Maximais Ternárias” a partir de “Grafos Ternários” (PPDI-S5_M6)

O modelo PPDI-S5_M5 fornece bons limitantes de PL, melhorando o *Branch-and-Bound* de *solvers* comerciais. Entretanto, ele apresenta ainda um grande número de restrições porque o número de restrições de adjacência é grande. Entretanto, este número pode ser reduzido se as “cliques maximais ternárias” forem obtidas dos “grafos ternários”.

Um “grafo ternário” $G=(V,A)$ é criado considerando todas as restrições ternárias. Nesse grafo, cada variável binária do modelo de Vasquez e Hao (2001) representa um vértice e suas arestas são criadas a partir das restrições ternárias. Agora uma “clique ternária” pode ser obtida para cada um desses vértices, analisando sua vizinhança.

Considerando novamente o exemplo da Figura 6.6. Aplicando o algoritmo para encontrar as “cliques maximais ternárias” descrito na seção anterior, as seguintes restrições válidas são encontradas: $x_i + x_j + x_k + x_l \leq 2$, $x_i + x_j + x_k + x_m \leq 2$, $x_i + x_j + x_k + x_n \leq 2$,

$x_i + x_j + x_l + x_m \leq 2$, $x_i + x_j + x_l + x_n \leq 2$ e $x_i + x_j + x_n + x_m \leq 2$. Note que a restrição válida $x_i + x_j + x_k + x_l + x_m + x_n \leq 2$ descrita na figura não é obtida. Sendo assim, o algoritmo para encontrar “cliques ternárias maximais” baseado em restrições binárias, apresenta problemas. Uma forma de contornar esse problema é trabalhar com o “grafo ternário”.

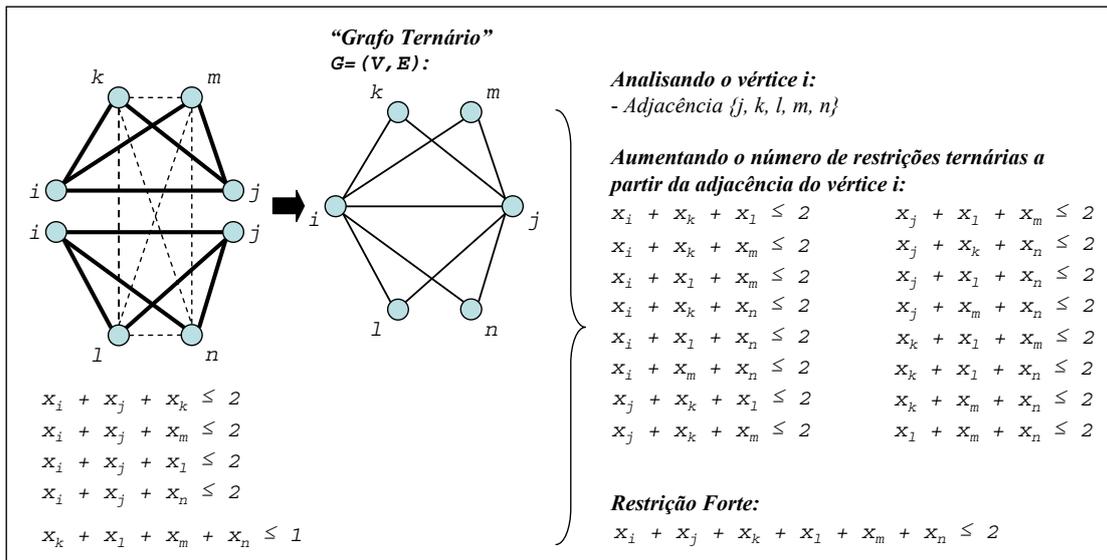


FIGURA 6.9 – Inequação válida para o PPDI-S5 usando “cliques maximais ternárias” obtidas a partir de “grafos ternários”.

A Figura 6.9 mostra o “grafo ternário” do exemplo descrito na Figura 6.6. Observe que, aplicando o processo descrito na seção anterior para aumentar o número de restrições ternárias próximas do vértice i , pode-se deduzir a mesma inequação válida mostrada na Figura 6.6 que é uma “clique maximal ternária” envolvendo os vértices x_i, x_j, x_k, x_l, x_m e x_n .

Seja Ψ agora um conjunto com as restrições de “cliques maximais ternárias” conforme descrito na Figura 6.9. Para encontrar este conjunto, pode-se aplicar o algoritmo descrito na Figura 6.10.

Observe que o conjunto de “cliques maximais ternárias” definido com o algoritmo da Figura 6.10 contém todas as restrições de par-binários e as ternárias com cliques dos modelos PPDI-S5_M3 e PPDI-S5_M4.

A seguir é apresentando o último modelo proposto, denominado aqui de PPDI-S5_M6:

$$v(PPDI - S5_M_6) = Max \left\{ \sum_{i=1}^m b_i x_i \right\} \quad (6.48)$$

(PPDIS-S5_M6) Sujeito a:

$$\sum_{i=1}^m c_i x_i \leq Cap \quad (6.49)$$

$$\sum_{i \in k} x_i \leq 1 \quad \forall \text{Clique Maximal } k = 1, \dots, K \quad (6.50)$$

$$\sum_{i \in \psi} x_i \leq 2 \quad \forall \text{Clique Maximal Ternária} \quad (6.51)$$

(obtida a partir do Grafo Ternário) $\psi = 1, \dots, \Psi$

$$x_i \in \{0, 1\} \quad \forall i = 1 \dots m \quad (6.52)$$

```

Algoritmo Gera_Cliq_Max_Ter_de_Grafos_Ternários()
// Seja:
// -  $\Psi$  um conjunto de "cliques maximais ternárias"
// - S um conjunto de variáveis
// -  $G = (V, E)$  um "grafo ternário"

1   $\Psi \leftarrow C_3$ 
2  Crie  $G = (V, E)$ 
3  Para cada  $x_i \in V$  Faça
4     $S \leftarrow \emptyset$ 
5    Selecione todas as variáveis  $x_j$  na vizinhança de  $x_i$  e as
      adicione em S
6    Com  $x_i$  e sua vizinhança S, gere todas as "cliques
      ternárias" que tem  $x_i$  como vértice e as adicione em  $\Psi$ 
7  Fim Para
8  Remova todas as "cliques ternárias" de  $\Psi$  que estão dominadas

```

FIGURA 6.10 – Algoritmo para gerar “cliques maximais ternárias” a partir de “grafos ternários”.

O modelo S5-DPSP_M5 tem um certo conjunto de “cliques maximal ternárias” enquanto o modelo S5-DPSP_M6 tem um conjunto menor, porém dominante.

6.5 Resultados Computacionais

Nesta seção são apresentados os resultados computacionais das relaxações lagrangeanas e dos novos modelos propostos. Foram utilizadas as instâncias propostas por Bensana et al. (1999).

O algoritmo de subgradientes implementado novamente foi o proposto por Narciso e Lorena (1999). A máquina utilizada foi um Pentium IV 3,33 GHz com 1,0 GB de Memória RAM.

O particionamento do grafo para a LagClus foi novamente realizado com o METIS. Os multiplicadores lagrangeanos foram inicializados com zero nas duas relaxações L_λ PPDIS-S5 e LG_λ PPDIS-S5. Além disso, os passos foram atualizados como realizado no PRCP, PCPP e PEUC, porém não foi implementada uma heurística lagrangeana, ao invés disso, as melhores soluções conhecidas propostas por Vasquez e Hao (2001) foram utilizadas como melhores limitantes inferiores. Os critérios de parada foram os mesmos adotados no PCPP e PEUC.

Para resolver as relaxações utilizou-se o CPLEX 7.5 enquanto nos testes das novas formulações foi utilizado o CPLEX 10.

6.5.1 Resultados das Relaxações Lagrangeanas

A Tabela 6.1 apresenta os resultados da LagClus e da relaxação de PL do modelo de Vasquez e Hao (2001). As colunas referem-se:

- Prob. - Instância considerada;
- *Solução Ótima*/Melhor - Indica a melhor solução conhecida reportada por Vasquez e Hao (2001) ou a solução ótima da instância;
- UB_{PL} - Valor do limitante superior fornecido pela relaxação de PL;
- T_{PL} - Indica o tempo utilizado pelo CPLEX 7.5 para resolver a relaxação de PL;
- GAP (%) - O *gap* percentual entre os limitantes superior e inferior calculado como: $GAP = \frac{(UB-LB)}{UB} \times 100$;
- # de Clusters - Número de *clusters* considerado no particionamento para a LagClus;

- $UB_{LagClus}$ - Valor do limitante superior fornecido pela LagClus;
- Iter. - Número de iterações realizadas pela LagClus; e
- $T_{LagClus}$ - Tempo utilizado pela LagClus até encontrar uma das condições de parada.

TABELA 6.1 – Resultados das relaxações de PL e da LagClus.

	Prob.	Solução Ótima/ Melhor (LB)	PPDI-S5 com o CPLEX 7.5			LagClus				
			UB_{PL}	T_{PL}	GAP (%)	# de Clusters	$UB_{LagClus}$	Iter.	$T_{LagClus}$ (s)	GAP (%)
Instâncias sem restrição de capacidade	54	70	83,00	0,02	15,66	2	70	1	0	0,00
	29	12032	13057,00	0,00	7,85	2	12032	1	0	0,00
	42	108067	190567,50	0,03	43,29	2	108067	2	0	0,00
	28	56053	221090,50	0,08	74,65	2	64742,25	145	9	13,42
	5	115	315,00	0,11	63,49	2	115	1	3	0,00
	404	49	96,00	0,00	48,96	2	49,99	3	0	0,00
	408	3082	5188,00	0,05	40,59	2	3082,95	5	2	0,00
	412	16102	31323,50	0,05	48,59	2	16103,23	145	86	0,01
	11	22120	40416,00	0,17	45,27	2	22121,81	145	163	0,01
	503	9096	12637,50	0,01	28,02	2	9096	2	0	0,00
	505	13100	22236,00	0,06	41,09	2	13100	2	0	0,00
	507	15137	27361,50	0,08	44,68	2	15138	145	145	0,01
509	19125	36394,00	0,16	47,45	2	19125	2	4	0,00	
Instâncias com capacidade	1401	176056	300000,00	0,02	41,31	5	211968,50	145	49	16,94
	1403	176140	300149,00	0,51	41,32	5	207145,22	145	165	14,97
	1405	176179	300207,00	3,39	41,31	8	205483,12	145	15929	14,26
	1021	176246	300385,00	3,45	41,33	5	194909,72	145	9612	9,58
	1502	61158	64160,50	0,01	4,68	2	61158	2	0	0,00
	1504	124243	191279,00	0,17	35,05	3	124243,57	12	1168	0,00
	1506	168247	276863,00	1,48	39,23	5	193788,21	145	976	13,18

Note que a LagClus forneceu limitantes melhores que a relaxação de PL e provou a solução ótima de algumas instâncias. Em especial, ela provou a solução ótima da instância 1504, pois a diferença entre o limitante superior e inferior foi menor que 1. Essa solução ótima ainda não era conhecida.

Tentando reduzir os *gaps* das instâncias com restrições de capacidade, novos experimentos foram realizados com um grafo ponderado. As arestas do grafo de conflitos definido da Seção 6.3.1 receberam pesos conforme suas relações com restrições binárias e ternárias. O peso $\omega(x_i, x_j)$ associado a uma aresta (x_i, x_j) inicia com zero e:

- Para cada restrição ternária que contenha x_i e x_j , o respectivo peso deve ser atualizado da seguinte maneira: $\omega(x_i, x_j) \leftarrow \omega(x_i, x_j) + 1$;
- Se x_i e x_j aparecem em ao menos uma restrição de adjacência definida por 6.2 ou 6.3, o peso é atualizado como $\omega(x_i, x_j) \leftarrow \omega(x_i, x_j) + 4$.

O procedimento acima é semelhante ao proposto por Vasquez e Hao (2003), porém os autores utilizaram um número pequeno de *clusters*, o que gerou um bom limitante.

Sendo assim, o grafo G foi particionado novamente pelo METIS em \bar{P} *clusters*. Como agora o grafo possui pesos, o METIS busca definir conjuntos de vértices formados por aproximadamente $|V|/\bar{P}$ elementos tal que a soma dos pesos das arestas que conectam os vértices em diferentes partições, seja minimizada. A LagClus com o grafo ponderado será aqui denominada LagClus-Peso e seus resultados estão mostrados na Tabela 6.2 a seguir.

TABELA 6.2 – Resultados da LagClus-Peso.

	Prob.	Solução Ótima/ Melhor (LB)	LagClus				GAP (%)
			# de Clusters	UB _{LagClus-Peso}	Iter.	T _{LagClus-Peso} (s)	
Instâncias sem restrição de capacidade	54	70	2	70	1	0	0,00
	29	12032	2	12032	1	0	0,00
	42	108067	2	108067	2	0	0,00
	28	56053	2	62053	145	66	9,67
	5	115	2	115	1	2	0,00
	404	49	2	49,99	3	0	0,00
	408	3082	2	3082,89	5	2	0,00
	412	16102	2	16103	145	93	0,01
	11	22120	2	22122,27	145	144	0,01
	503	9096	2	9096	2	0	0,00
	505	13100	2	13100	2	0	0,00
	507	15137	2	15138	145	110	0,01
509	19125	2	19125	2	3	0,00	
Instâncias com capacidade	1401	176056	5	192530,85	145	423	8,56
	1403	176140	5	195654,93	145	6888	9,97
	1405	176179	8	198952,08	145	6489	11,45
	1021	176246	5	189080,24	145	53955	6,79
	1502	61158	2	61158	2	0	0,00
	1504	124243	3	124243,62	10	904	0,00
	1506	168247	5	179699,50	145	32280	6,37

Considerando a Tabela 6.2, verifica-se que os *gaps* reduziram, mas os tempos computacionais aumentaram para algumas instâncias (veja as instâncias 1401, 1403, 1405, 1021 e 1506). O tempo computacional reduziu para as instâncias 1405 e 1504.

A Tabela 6.3 apresenta os resultados da relaxação L_λ PPDI-S5. Note que os *gaps* encontrados para as instâncias com restrição de capacidade reduziram. Entretanto, o tempo computacional foi maior e para as instâncias 1504 e 1506 o processo não foi finalizado após um 24h de processamento.

TABELA 6.3 – Resultados com L_{λ} PPDIS-S5.

	Prob.	Solução Ótima/ Melhor (LB)	L_{λ} PPDI-S5			
			$UB_{L_{\lambda}PPDI-S5}$	Iter.	$T_{L_{\lambda}PPDI-S5}$ (s)	GAP (%)
Instâncias sem restrição de capacidade	54	70	70	1	0,02	0,00
	29	12032	12032	1	0,02	0,00
	42	108067	108067	1	0,05	0,00
	28	56053	56053	1	0,10	0,00
	5	115	115	1	1,25	0,00
	404	49	49	2	0,01	0,00
	408	3082	3085	145	10,45	0,09
	412	16102	16105	145	24,23	0,02
	11	22120	22123	145	38,30	0,01
	503	9096	9098	145	3,23	0,02
	505	13100	13102	145	8,02	0,02
	507	15137	15139	145	37,24	0,01
	509	19125	19128	145	55,25	0,02
Instâncias com capacidade	1401	176056	184783,23	145	4811	4,72
	1403	176140	184933,62	145	56458	4,76
	1405	176179	185007,84	145	1322	4,77
	1021	176246	185112,74	145	11573	4,79
	1502	61158	61158	0	0	0,00
	1504	124243	-	-	-	-
	1506	168247	-	-	-	-

6.5.2 Resultados com os Novos Modelos para o PPDI-S5

A Tabela 6.4 apresenta o número total de imagens, variáveis binárias e restrições geradas por cada um dos modelos propostos. As colunas são auto-explicativas com exceção da coluna “# de Imagens” que representa o número de imagens presentes no problema.

Considerando os valores médios, verifica-se que o modelo PPDI-S5_M₁ gera o menor número de restrições e o modelo PPDI-S5_M₃ o maior. O modelo PPDI-S5_M₃ inclusive gera mais restrições que o modelo base de Vasquez e Hao (2001).

Os testes computacionais foram realizados considerando cinco cenários diferentes:

- 1^o. Limitantes de PL fornecidos pelos modelos;
- 2^o. CPLEX 10 trabalhando até encontrar 0,01% de *gap*;
- 3^o. CPLEX 10 trabalhando por 2 horas (7200s) no máximo;
- 4^o. CPLEX 10 trabalhando até encontrar a solução ótima ou parar por falta de memória; e
- 5^o. CPLEX 10 sem o seu método de geração de cortes próprio, trabalhando por até 2 horas (7200s).

O modelo PPDI-S5_M₁ não foi utilizado para comparação uma vez que seus resultados foram bem inferiores aos do modelo original de Vasquez e Hao (2001).

Na Tabela 6.5 pode-se comparar os limitantes de PL gerados pelos novos modelos (de PPDI-S5_M₂ a PPDI-S5_M₆). Os *gaps* gerados pelos modelos propostos são menores que os produzidos pelo modelo de Vasquez e Hao (2001). Além disso, pode-se perceber que o modelo S5-DPSP_M₆ forneceu os melhores limitantes em um tempo computacional aceitável.

TABELA 6.4 – Número de restrições geradas por cada modelo.

	Prob.	# de Imagens	<i>m</i>	PPDI-S5	PPDI-S5_M ₁	PPDI-S5_M ₂	PPDI-S5_M ₃	PPDI-S5_M ₄	PPDI-S5_M ₅	PPDI-S5_M ₆
Instâncias sem restrição de capacidade	54	67	125	441	169	228	277	223	227	212
	29	82	120	629	147	323	323	323	323	323
	42	190	304	1883	489	1043	1223	1029	1035	1029
	28	230	346	6950	1133	3939	8372	3971	3976	3521
	5	309	809	14599	1748	10028	12896	10376	10388	9805
	404	100	158	966	220	1267	408	363	364	353
	408	200	328	3013	865	1643	2990	1727	1744	1429
	412	300	544	7096	1256	3465	4812	3549	3566	3251
	11	364	692	14339	5828	9050	67840	12315	12352	5589
	503	143	259	849	357	362	459	358	368	317
	505	240	448	3296	1087	1327	3502	1257	1277	961
	507	311	573	7969	3170	4429	24402	5113	5145	2772
	509	348	652	12047	4954	7433	51657	9082	9115	4546
Instâncias com capacidade	1401	488	914	15020	4385	10553	37112	12017	12047	8393
	1403	665	1317	19198	5980	12795	46967	15325	15373	10050
	1405	855	1815	29547	7752	19848	61005	22880	22947	16571
	1021	1057	2355	36583	9806	23908	76698	27737	27827	19752
	1502	209	413	428	324	328	332	332	338	311
	1504	605	1253	6313	2392	2607	6522	2792	2826	2034
	1506	940	2060	24369	8120	13319	53645	16085	16165	9902
Média	385,15	774,25	10276,75	3009,10	6394,75	23072,10	7342,70	7370,15	5056,05	

Na Tabela 6.6 são mostrados os resultados para o segundo cenário, ou seja, com o CPLEX 10 executando até encontrar um gap de 0,01%. Nesse caso, LB e UB são os limitantes inferior e superior, respectivamente, fornecidos pelo CPLEX. A coluna GAP é calculada da seguinte maneira: $GAP = \frac{(UB-LB)}{UB} \times 100$.

Os resultados para as instâncias sem restrição de capacidade não mudaram significativamente de um modelo para outro. Elas podem ser consideradas “fáceis” para todos os modelos, inclusive para o PPDI-S5, e por isso não serão aqui mostradas.

Considerando a Tabela 6.6, o CPLEX 10 teve dificuldades para encontrar 0,01% de *gap* com o modelo PPDI-S5_M₂ mesmo trabalhando por mais de 15 horas como será visto na Tabela 6.8. O CPLEX com o modelo PPDI-S5_M₃ forneceu melhores resultados do que com o modelo PPDI-S5_M₂ porém quando comparado com os do modelo PPDI-S5, os resultados não foram bons. Isso indica que os cortes embutidos no CPLEX são mais eficientes que os cortes contidos em PPDI-S5_M₃.

TABELA 6.5 – Resultados da relaxação de PL.

	Prob.	Solução Ótima/ Melhor (LB)	PPDI-S5			PPDI-S5_M ₂			PPDI-S5_M ₃		
			PL	T (s)	GAP (%)	PL	T (s)	GAP (%)	PL	T (s)	GAP (%)
Instâncias sem restrição de capacidade	54	70	83,00	0,02	15,66	71,17	0,02	1,64	70,58	0,02	0,82
	29	12032	13057,00	0,00	7,85	12032	0,02	0,00	12032	0,00	0,00
	42	108067	190567,50	0,03	43,29	108067	0,03	0,00	108067	0,05	0,00
	28	56053	221090,50	0,08	74,65	56853	0,25	1,41	56853	0,38	1,41
	5	115	315,00	0,11	63,49	115,50	0,80	0,43	115,50	0,86	0,43
	404	49	96,00	0,00	48,96	49,67	0,00	1,35	49,33	0,00	0,67
	408	3082	5188,00	0,05	40,59	3084,67	0,03	0,09	3083,83	0,11	0,06
	412	16102	31323,50	0,05	48,59	17092,67	0,11	5,80	17091,83	0,13	5,79
	11	22120	40416,00	0,17	45,27	23110,67	0,69	4,29	23109,83	1,86	4,28
	503	9096	12637,50	0,01	28,02	9592,50	0,00	5,18	9592	0,00	5,17
	505	13100	22236,00	0,06	41,09	14098	0,03	7,08	14097,50	0,05	7,08
507	15137	27361,50	0,08	44,68	16624,50	0,09	8,95	16624,50	0,70	8,95	
509	19125	36394,00	0,16	47,45	20698,50	0,13	7,60	20697,67	1,63	7,60	
Instâncias com capacidade	1401	176056	300000,00	0,02	41,31	184753,54	2,92	4,71	183909,95	5,88	4,27
	1403	176140	300149,00	0,51	41,32	184840,54	1,70	4,71	183996,95	4,30	4,27
	1405	176179	300207,00	3,39	41,31	184877,64	3,24	4,71	184033,31	7,63	4,27
	1021	176246	300385,00	3,45	41,33	184945,64	5,17	4,70	184101,31	11,05	4,27
	1502	61158	64160,50	0,01	4,68	61158,00	0,01	0,00	61158,00	0,00	0,00
	1504	124243	191279,00	0,17	35,05	126245,93	0,16	1,59	126245,76	0,53	1,59
	1506	168247	276863,00	1,48	39,23	172892,51	3,23	2,69	172892,51	3,95	2,69

...Continuação...

	Prob.	Solução Ótima/ Melhor (LB)	PPDI-S5_M ₄			PPDI-S5_M ₅			PPDI-S5_M ₆		
			PL	T (s)	GAP (%)	PL	T (s)	GAP (%)	PL	T (s)	GAP (%)
Instâncias sem restrição de capacidade	54	70	70	0,00	0,00	70	0,00	0,00	70	0,00	0,00
	29	12032	12032	0,00	0,00	12032	0,00	0,00	12032	0,00	0,00
	42	108067	108067	0,03	0,00	108067	0,03	0,00	108067	0,03	0,01
	28	56053	56853	0,30	1,41	56853	0,20	1,41	56853	0,23	1,41
	5	115	115,50	0,94	0,43	115,50	0,94	0,43	115,50	0,94	0,43
	404	49	49	0,00	0,00	49	0,00	0,00	49	0,00	0,00
	408	3082	3083,50	0,05	0,05	3083	0,05	0,03	3083	0,05	0,03
	412	16102	17091,50	0,17	5,79	17091	0,11	5,79	17091	0,09	5,79
	11	22120	23109,50	0,45	4,28	23109	0,38	4,28	23107	0,14	4,27
	503	9096	9591,67	0,00	5,17	9591	0,02	5,16	9591	0,00	5,16
	505	13100	14097,50	0,01	7,08	14097	0,03	7,07	14097	0,02	7,07
507	15137	16623,84	0,37	8,94	16623,50	0,11	8,94	16623,50	0,06	8,94	
509	19125	20697,33	0,24	7,60	20696,50	0,22	7,59	20696,50	0,17	7,59	
Instâncias com capacidade	1401	176056	183325,18	3,52	3,97	182321,31	1,80	3,44	180045,70	1,30	2,22
	1403	176140	183412,18	6,47	3,96	181019,79	2,22	2,70	180131,87	2,64	2,22
	1405	176179	183449,13	8,56	3,96	180981,66	4,30	2,65	180168,84	4,09	2,21
	1021	176246	183517,13	9,03	3,96	181049,66	7,58	2,65	180236,63	5,70	2,21
	1502	61158	61158,00	0,00	0,00	61158	0,00	0,00	61158	0,00	0,00
	1504	124243	126244,81	0,33	1,59	125618,94	0,17	1,10	125132,49	0,09	0,71
	1506	168247	172781,09	4,45	2,62	170448,19	4,94	1,29	169589,37	2,64	0,79

TABELA 6.6 – Resultados computacionais para um 0,01% de *gap*.

	Prob.	L _λ PPDI-S5				PPDI-S5_M ₂				PPDI-S5_M ₃			
		LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)
Instâncias com capacidade	1401	176056	176080,07	2889,55	0,01	176056	176073,05	1713,22	0,01	176056	176072,01	6464,48	0,009
	1403	176141	176165,65	6394,28	0,01	176141	176151,00	3624,89	0,01	176141	176147,79	9004,30	0,004
	1405	176179	176195,74	4051,56	0,01	176179	176189,17	5990,22	0,01	176179	176195,90	9454,02	0,010
	1021	176246	176263,54	9281,53	0,01	*	*	*	*	176246	176263,51	24224,78	0,010
	1502	61158	61158	0,02	0,00	61158	61158,00	0,02	0,00	61158	61158	0,02	0,000
	1504	124241	124252,33	11,13	0,01	124240	124248,00	4,88	0,01	124241	124247,00	10,56	0,005
	1506	168246	168262,00	2040,11	0,01	168246	168256,69	367,33	0,01	168246	168259,33	332,11	0,008
...Continuação...													
	Prob.	L _λ PPDI-S5_M ₄				PPDI-S5_M ₅				PPDI-S5_M ₆			
		LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)
Instâncias com capacidade	1401	176056	176057,08	978,99	0,001	176056	176059,78	325,72	0,002	176056	176056	47,92	0,000
	1403	176141	176146,50	1492,27	0,003	176141	176148,00	281,94	0,004	176141	176142,60	47,50	0,001
	1405	176178	176192,84	1914,23	0,008	176179	176192,59	446,81	0,008	176179	176182,69	84,86	0,002
	1021	176246	176259,31	4295,86	0,008	176246	176263,43	785,86	0,010	176246	176250,79	169,44	0,003
	1502	61158	61158	0,03	0,000	61158	61158	0,01	0,000	61158	61158	0,01	0,000
	1504	124241	124246,00	3,92	0,004	124243	124245,33	2,03	0,002	124239	124244,54	0,27	0,004
	1506	168247	168252,92	187,92	0,004	168247	168251,00	30,61	0,002	168244	168248,76	4,47	0,003

TABELA 6.7 – Resultados computacionais com o CPLEX trabalhando no máximo 7200s.

	Prob.	L _λ PPDI-S5				PPDI-S5_M ₂				PPDI-S5_M ₃			
		LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)
Instâncias com capacidade	1401	*	*	*	*	*	*	*	*	176056	176059,95	7200	0,002
	1403	*	*	*	*	*	*	*	*	176141	177042,15	7200	0,509
	1405	176179	176183,04	7200	0,002	176179	176183,05	7200	0,002	176179	176857,67	7200	0,384
	1021	176246	176848,32	7200	0,341	176244	179698,09	7200	1,922	176245	178248,50	7200	1,124
	1502	*	*	*	*	*	*	*	*	*	*	*	*
	1504	*	*	*	*	*	*	*	*	*	*	*	*
	1506	168247	168252,18	7200	0,003	168247	168251,30	7200	0,003	168247	168251,60	7200	0,003
...Continuação...													
	Prob.	L _λ PPDI-S5_M ₄				PPDI-S5_M ₅				PPDI-S5_M ₆			
		LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)
Instâncias com capacidade	1401	*	*	*	*	*	*	*	*	*	*	*	*
	1403	*	*	*	*	*	*	*	*	*	*	*	*
	1405	176179	176180,83	7200	0,001	*	*	*	*	*	*	*	*
	1021	176246	176249,55	7200	0,002	176246	176247	7200	0,001	*	*	*	*
	1502	*	*	*	*	*	*	*	*	*	*	*	*
	1504	*	*	*	*	*	*	*	*	*	*	*	*
	1506	168247	168248,44	7200	0,001	*	*	*	*	*	*	*	*

Agora o CPLEX com os modelos PPDI-S5_M₄, PPDI-S5_M₅ e PPDI-S5_M₆ encontrou *gaps* de 0,01% em um tempo computacional menor do que os outros modelos. O último modelo em especial (PPDI-S5_M₆) levou um tempo bem menor quando comparado com PPDI-S5_M₅. Note que a melhor solução conhecida para a instância 1403 foi alterada de 176140 para 176141.

Os resultados para o terceiro cenário estão mostrados na Tabela 6.7. Neste cenário o CPLEX 10 foi utilizado durante 7200s e neste caso, com as formulações PPDI-S5, PPDI-S5_M₂, PPDI-S5_M₄ e PPDI-S5_M₆, o CPLEX 10 resolveu de forma ótima as instâncias 1401, 1403, 1502 e 1504. Agora note que com o modelo PPDI-S5_M₅, ele

resolveu todas as instâncias, exceto a instância 1021 que permaneceu com um *gap* pequeno ao final do tempo estipulado. Por outro lado, repare que com o modelo PPDI-S5_M6 o CPLEX 10 resolveu todas as instâncias em um tempo inferior a 7200s.

TABELA 6.8 – Resultados computacionais do CPLEX até resolver o problema ou parar por falta de memória.

	Prob.	L ₁ PPDI-S5				PPDI-S5_M ₂				PPDI-S5_M ₃			
		LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)
Instâncias com capacidade	1401	176056	176056	2882,80	0,000	176056	176056,00	2054,81	0,000	176056	176056	8877,05	0,000
	1403	176141	176141	6612,80	0,000	176141	176141,00	6768,97	0,000	176141	176141	12887,39	0,000
	1405	176179	176182,70	10604,44	0,002	176179	176182,00	54524,61	0,002	176179	176182,30	264125,21	0,002
	1021	176246	176250,20	27609,20	0,002	176245	178664,08	17333,98	1,354	176246	176250,78	73930,70	0,003
	1502	61158	61158	0,02	0,000	61158	61158,00	0,02	0,000	61158	61158	0,02	0,000
	1504	124243	124243	305,16	0,000	124243	124243,00	163,52	0,000	124243	124243	86,59	0,000
	1506	168247	168252,00	12595,28	0,003	168247	168250,83	16181,59	0,002	168247	168250,65	67995,58	0,002
...Continuação...													
	Prob.	L ₁ PPDI-S5_M ₄				PPDI-S5_M ₅				PPDI-S5_M ₆			
		LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)
Instâncias com capacidade	1401	176056	176056	991,75	0,000	176056	176056	346,44	0,000	176056	176056	47,92	0,000
	1403	176141	176141	2104,59	0,000	176141	176141	322,17	0,000	176141	176141	47,59	0,000
	1405	176179	176179	13445,12	0,000	176179	176179	1153,66	0,000	176179	176179	95,34	0,000
	1021	176246	176246	231189,88	0,000	176246	176246	8317,36	0,000	176246	176246	192,49	0,000
	1502	61158	61158	0,03	0,000	61158	61158	0,01	0,000	61158	61158	0,01	0,000
	1504	124243	124243	13,7	0,000	124243	124243	3,45	0,000	124243	124243	0,53	0,000
	1506	168247	168247	76021,55	0,000	168247	168247	4743,25	0,000	168247	168247	41,81	0,000

A Tabela 6.8 apresenta os resultados encontrados para o quarto cenário, ou seja, com o CPLEX trabalhando até resolver de forma exata o problema ou parar por falta de memória.

As soluções ótimas das instâncias 1405, 1021 e 1506 não são encontradas com os modelos PPDI-S5, PPDI-S5_M2 e PPDI-S5_M3, mesmo trabalhando por mais de 3 dias (veja o tempo computacional da formulação PPDI-S5_M3 para a instância 1405).

Considerando agora o modelo PPDI-S5_M4, o CPLEX encontrou a solução ótima para todas as instâncias: rapidamente para as instâncias 1502 e 1504, e levando muito tempo nas demais.

Com o modelo PPDI-S5_M5 novamente o CPLEX encontrou, rapidamente, a solução ótima das instâncias 1502 e 1504. Porém também levou muito tempo para resolver as instâncias 1021 e 1506. Os melhores resultados foram obtidos com o modelo PPDI-S5_M6. Com esse modelo, o CPLEX encontrou todas as soluções ótimas em um tempo computacional reduzido se comparado com os demais modelos. No pior caso, o CPLEX levou aproximadamente 3 minutos para resolver a instância 1021.

Comparando agora as formulações PPDI-S5_M5 e PPDI-S5_M6; e levando em consideração o número total de restrições mostradas na Tabela 6.4, o modelo PPDI-S5_M6 tem em média 2313 restrições a menos que o modelo PPDI-S5_M5,

entretanto isso é suficiente para que ele seja mais eficiente.

A Tabela 6.9 apresenta os resultados do último cenário. Neste caso a geração de cortes do CPLEX foi removida e o tempo limitado em 7200s. Pela tabela percebe-se que o CPLEX com o modelo PPDI-S5_M6, mesmo sem os seus cortes, resolve todos os problemas em um tempo computacional aceitável. No pior caso, ele levou aproximadamente 12 minutos para resolver a maior instância. Outro fato interessante foi que o CPLEX resolveu a instância 1506 em um tempo computacional menor do que com os seus cortes (38,92 contra 41,81s).

TABELA 6.9 – CPLEX sem a sua geração própria de cortes.

	Prob.	L _λ PPDI-S5				PPDI-S5_M ₂				PPDI-S5_M ₃			
		LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)
Instâncias com capacidade	1401	176056	181687,03	7200	3,099	176056	176056	1745,73	0,000	176056	176056	5020,88	0,000
	1403	176140	181212,99	7200	2,799	176141	176143,50	7200	0,001	176141	176144,21	7200	0,002
	1405	176178	181274,61	7200	2,812	176179	178184,06	7200	1,125	176179	177679,78	7200	0,845
	1021	176244	181573,81	7200	2,935	176242	178696,20	7200	1,373	176244	177849,33	7200	0,903
	1502	61158	61158	0,22	0,000	61158	61158	0,02	0,000	61158	61158	0,02	0,000
	1504	124243	124244,20	7200	0,001	124243	124243	124,80	0,000	124243	124243	123,14	0,000
	1506	168247	168257,99	7200	0,007	168247	168251,67	7200	0,003	168247	168252,22	7200	0,003
...Continuação...													
	Prob.	L _λ PPDI-S5_M ₄				PPDI-S5_M ₅				PPDI-S5_M ₆			
		LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)	LB	UB	T (s)	GAP (%)
Instâncias com capacidade	1401	176056	176056	919,84	0,000	176056	176056	284,75	0,000	176056	176056	58,91	0,000
	1403	176141	176141	1356,08	0,000	176141	176141	622,03	0,000	176041	176041	62,83	0,000
	1405	176179	176183,08	7200	0,002	176179	176179	4565,31	0,000	176179	176179	217,67	0,000
	1021	176246	176250,29	7200	0,002	176246	176247	7200	0,001	176246	176246	729,36	0,000
	1502	61158	61158	0,02	0,000	61158	61158	0,00	0,000	61158	61158	0,01	0,000
	1504	124243	124243	12,44	0,000	124243	124243	4,36	0,000	124243	124243	1,17	0,000
	1506	168247	168249,63	7200	0,002	168247	168247	2002,51	0,000	168247	168247	38,92	0,000

TABELA 6.10 – Comparação com a literatura.

	Prob.	Solução Ótima LB	Geração de Colunas (Gabrel, 1999)		(Gabrel, 2005)		Vasquez e Hao (2003)		Relaxações Lagrangeanas		PPDI-S5_M ₆	
			UB	GAP (%)	UB	GAP (%)	UB	GAP (%)	UB	GAP (%)	UB	GAP (%)
Instâncias sem restrição de capacidade	54	70	71,44	2,02	71	1,41	70	0,00	70	0,00	70	0,00
	29	12032	-	-	-	-	12032	0,00	12032	0,00	12032	0,00
	42	108067	108067,00	0,00	108067	0,00	108067	0,00	108067	0,00	108067	0,00
	28	56053	70053,00	19,98	67053	16,40	58053	3,45	56053	0,00	56053	0,00
	5	115	119,00	3,36	151	23,84	116	0,86	115	0,00	115	0,00
	404	49	53,66	8,68	49	0,00	49	0,00	49,99	0,00	49	0,00
	408	3082	3091,75	0,32	3094	0,39	3083	0,03	3082,89	0,00	3082	0,00
	412	16102	71,44	-22439,19	23580	31,71	16102	0,00	16103	0,01	16102	0,00
	11	22120	-	-	-	-	22120	0,00	22122,27	0,01	22120	0,00
	503	9096	9595,00	5,20	9601	5,26	9096	0,00	9096	0,00	9096	0,00
	505	13100	14107,75	7,14	15803	17,10	13103	0,02	13100	0,00	13100	0,00
	507	15137	17639,22	14,19	25278	40,12	15137	0,00	15138	0,01	15137	0,00
	509	19125	-	-	25684	25,54	19125	0,00	19125	0,00	19125	0,00
Instâncias com capacidade	1401	176056	-	-	188370	6,54	180062	2,22	184783,23	4,72	176056	0,00
	1403	176141	-	-	188467	6,54	180160	2,23	184933,62	4,76	176141	0,00
	1405	176179	-	-	188551	6,56	179226	1,70	185007,84	4,77	176179	0,00
	1021	176246	-	-	-	-	177304	0,60	185112,74	4,79	176246	0,00
	1502	61158	-	-	64155	4,67	61158	0,00	61158	0,00	61158	0,00
	1504	124243	-	-	139585	10,99	124258	0,01	124243,62	0,00	124243	0,00
	1506	168247	-	-	183978	8,55	168294	0,03	179699,50	6,37	168247	0,00

6.5.3 Comparação com a Literatura

A Tabela 6.10 permite realizar uma comparação dos *gaps* disponíveis na literatura com os principais obtidos aqui. Note que os melhores *gaps* haviam sido encontrados por Vasquez e Hao (2003) e agora todas as soluções ótimas foram encontradas.

Esses resultados mostram que a Busca Tabu de Vasquez e Hao (2001) produz boas soluções. Para as instâncias com restrição de capacidade, o algoritmo encontrou todas as soluções ótimas, exceto para a instância 1403.

6.6 Considerações Finais

Este capítulo apresentou o Problema da Programação Diária de Imagens (PPDI) de um satélite de observação terrestre. Em especial foi analisado o Satélite SPOT 5.

Um estudo sobre relaxações foi realizado e mesmo o problema não tendo um grafo de conflitos bem definido, a LagClus foi testada. Resultados computacionais mostraram que esta relaxação permite encontrar as soluções ótimas de algumas instâncias pequenas, e de duas instâncias maiores sendo que a solução ótima de uma delas não era conhecida até então.

Devido às limitações da LagClus, modelos alternativos baseados em inequações válidas foram estudados. Os dois últimos (PPDI-S5_M5 e PPDI-S5_M6) forneceram os melhores resultados, permitindo encontrar as soluções ótimas de todas as instâncias propostas por Bensana et al. (1999). Em especial, o último modelo necessita de um tempo computacional pequeno para encontrar todas as soluções ótimas.

Essas formulações foram obtidas usando os conceitos da teoria de cortes apresentados no Capítulo 3.

CAPÍTULO 7

CONCLUSÕES E TRABALHOS FUTUROS

A relaxação de um problema de Otimização Combinatória permite avaliar uma solução viável fornecida para o problema. As soluções das relaxações fornecem limitantes que ao serem comparados com o limitante de uma solução viável, permitem estimar se, por exemplo, a solução viável é ou não a solução ótima.

Existem várias relaxações comumente utilizadas na literatura: de Programação Linear, *Surrogate*, Lagrangeana e Lagrangeana/*Surrogate*. Dentre estas, destaca-se a relaxação lagrangeana que vem sendo utilizada largamente com sucesso em problemas de otimização. Esta relaxação utiliza o método dual de otimização por subgradientes que também é bem conhecido na literatura.

Por outro lado, a técnica tradicionalmente empregada na relaxação lagrangeana consiste em identificar e relaxar um subconjunto de restrições do problema, de tal maneira que o problema relaxado seja de “fácil” solução. Em seguida, ao realizar a otimização do dual lagrangeano por subgradientes, buscam-se melhores multiplicadores lagrangeanos.

Dependendo do problema, ao relaxar no sentido lagrangeano um subconjunto completo de restrições, a relaxação resultante pode não gerar bons limitantes e não propiciar também o desenvolvimento de uma heurística lagrangeana adequada.

Entretanto, alguns problemas de otimização combinatória podem ser representados por um grafo especial denominado grafo de conflitos. Quando estes grafos apresentam-se esparsos, bem adaptados para uma fase de particionamento de grafos, ou seja, quando apresentam agrupamentos de vértices bem definidos (*clusters*), as arestas que conectam os *clusters* podem ser relaxadas no sentido lagrangeano, e o problema relaxado pode ser decomposto e resolvido.

A estratégia acima possui uma vantagem em relação à relaxação lagrangeana tradicional por que não relaxa completamente um subconjunto de restrições e sim, parte dele. Essa idéia consiste na Relaxação Lagrangeana com *Clusters* (LagClus) proposta neste trabalho.

Ao decompor o problema original em *clusters*, pode-se resolvê-lo também usando um Algoritmo de Geração de Colunas em que o problema mestre restrito é formado pelas restrições de conexão (restrições de acoplamento), e os subproblemas geradores de colunas pelos *clusters* obtidos.

Além disso, este trabalho propôs também novos modelos matemáticos, baseados na teoria de cortes, para alguns problemas de otimização combinatória. Esses modelos utilizam tanto cortes tradicionais dispostos na literatura, quanto cortes específicos que dependem da natureza do problema tratado.

Sendo assim, as principais contribuições e as próximas etapas deste trabalho são apresentadas a seguir.

7.1 Resumos das Contribuições

As principais contribuições deste trabalho estão divididas em três linhas: Modelos Matemáticos, Relaxação Lagrangeana com *Clusters* e Algoritmo de Geração de Colunas que usa a idéia da divisão em *clusters* do grafo de conflitos. Com isso, a seguir são apresentadas as principais contribuições conforme as linhas descritas acima.

7.1.1 Modelos Matemáticos

Considerando o Problema da Rotulação Cartográfica de Pontos (PRCP), a abordagem do Problema da Minimização do Número de Conflitos (PMNC) foi apresentada juntamente com dois novos modelos: um baseado nas arestas do grafo de conflitos (PMNC^a) e outro baseado nos pontos a serem rotulados (PMNC^p). Este último apresenta um número menor de restrições.

O CPLEX, com os dois modelos acima, obteve rapidamente as soluções ótimas de problemas propostos na literatura com até 500 pontos. Porém, ele falhou em alguns problemas maiores com 750 pontos e em todas as instâncias testadas com 1000 pontos (Ribeiro e Lorena, 2004a, 2006c,a).

Considerando agora o Problema da Estivagem de Unidades de Celulose (PEUC), a formulação atualmente utilizada para o Problema do Carregamento de Paletes do Produtor (PCPP) pode ser utilizada, mesmo o PEUC sendo um problema tridimensional. Isto porque existem restrições operacionais que o reduzem para o caso bidimensional, e ao aplicar a tipologia de Dyckhoff (1990), o PEUC pode ser visto como um PCPP (Ribeiro e Lorena, 2006d). Consequentemente a formulação proposta por Beasley (1985) pode ser utilizada no PEUC.

Como o CPLEX apresenta problema com as instâncias do PCPP, com as do PEUC, esse problema é ainda mais agravado. Isto porque as instâncias do PEUC apresentam um maior grau de dificuldade.

Por último, alguns modelos matemáticos foram apresentados para o Problema da Programação Diária de Imagens do Satélite SPOT-5 (PPDI-S5). Esses modelos foram baseados na teoria de cortes (Ribeiro et al., 2007).

Em especial, a última formulação (PPDI-S5_M₆), que inclui cliques maximais e “cliques maximais ternárias”, possibilitou ao CPLEX encontrar as soluções ótimas de todas as instâncias com restrição de capacidade. Em especial, 6 novas soluções ótimas foram encontradas das 7 instâncias propostas na literatura. No pior caso, o CPLEX com a formulação PPDI-S5_M₆ foi capaz de encontrar a solução ótima da maior instância em aproximadamente 3 minutos, mostrando que esta formulação pode ser útil no dia a dia dos centros de estudos espaciais.

7.1.2 Relaxação Lagrangeana com *Clusters* (LagClus)

Foram apresentados resultados da LagClus para 4 problemas: PRCP, PCPP, PEUC e PPDI. Considerando o PRCP modelado com a nova abordagem proposta (PMNC), a LagClus apresentou um resultado bem superior ao do CPLEX. Ela provou as soluções ótimas de todas as instâncias com 100, 250, 500 e 750 pontos; e das 25 instâncias com 1000 pontos, 7 foram resolvidas de forma exata contra nenhuma do CPLEX. Isso demonstra que a LagClus é uma relaxação forte.

Ao comparar a qualidade da rotulação, ou seja, o percentual de rótulos livres, mesmo a abordagem do PMNC sendo inferior às outras duas, PRCP como um Problema do Máximo Conjunto Independente de Vértices (PMCIIV) e como um Problema do Máximo Número de Rótulos Sem Conflitos (PMNRSC), a LagClus apresentou resultados melhores que os presentes na literatura (Ribeiro e Lorena, 2006c,a).

Considerando agora os resultados da LagClus para o PCPP, novamente essa relaxação apresentou um comportamento interessante. Provou uma solução ótima de uma instância na literatura considerada difícil para uma relaxação lagrangeana, e encontrou outras soluções ótimas de instâncias com até 100 caixas. No pior caso, o limitante da LagClus ficou a uma caixa da solução ótima (Ribeiro e Lorena, 2007b).

Dado o comportamento descrito acima, a LagClus foi aplicada no PEUC e novamente os resultados foram interessantes. O limitante da LagClus, quando comparado aos limitantes geométricos presentes na literatura, foi melhor na maioria das instâncias. Além disso, a heurística lagrangeana apresentou padrões de estivagem melhores que os praticados nos portos brasileiros (Ribeiro e Lorena, 2006d), propiciando a redução de custos logísticos.

A LagClus também foi aplicada no PPDI-S5 mas seus resultados não foram tão

interessantes. Acredita-se que isso aconteceu porque o grafo definido para o problema não é um grafo de conflitos. Existem muitas restrições ternárias na formulação do PPDI-S5 que não podem ser representadas por restrições de adjacência. Mesmo assim, a LagClus provou a solução ótima, ainda não conhecida, de uma instância proposta na literatura.

7.1.3 Algoritmo de Geração de Colunas

A decomposição realizada pela LagClus permite a solução do problema pela técnica de geração de colunas. O Algoritmo de Geração de Colunas (AGC) foi aplicado no PRCP e PCPP (Ribeiro e Lorena, 2007b,a).

No caso do PRCP como um PMNC, o AGC apresentou resultados melhores que os da LagClus. Acredita-se que este fato tenha ocorrido devido à diversidade do conjunto inicial de colunas. Ao dividir uma solução viável nos *clusters*, estas podem ser combinadas com as novas colunas geradas durante o processo de geração, obtendo um bom limitante de PL. Além disso, como mostrado, existe uma equivalência entre o dual lagrangeano e a decomposição de Dantzig-Wolfe, com isso, o processo de geração de colunas pode ser finalizado quando o limitante de PL do PMR atinge o limitante da LagClus.

Considerando o PRCP, o AGC encontrou a solução ótima de todas as instâncias com 100, 250, 500 e 750 pontos. Das 25 instâncias propostas na literatura com 1000 pontos, o AGC encontrou 10 soluções ótimas contra 7 da LagClus e nenhuma do CPLEX. Além disso, o tempo total do AGC foi bem inferior ao da LagClus e do CPLEX.

Ao ser aplicado sobre as 10 instâncias do PCPP consideradas difíceis para uma relaxação lagrangeana, novamente o AGC apresentou bons resultados ao encontrar todas as soluções ótimas. Além disso, resultados computacionais mostraram a equivalência entre o AGC e a LagClus como mostrado na Figura 5.10.

Esse algoritmo quando aplicado em instâncias maiores do PCPP e PEUC, apresentou uma convergência lenta, indicando que estratégias de eliminação de colunas improdutivas devem ser incorporadas. Outra questão importante a ser destacada está relacionada ao conjunto inicial de colunas que foi gerado por uma heurística simples. Ao considerar, nesse conjunto inicial soluções de heurísticas mais elaboradas como as heurísticas de blocos, acredita-se que a convergência melhore.

7.2 Trabalhos Futuros

A relaxação apresentada neste trabalho aponta para algumas direções de pesquisa. Embora os resultados obtidos tenham sido satisfatórios para problemas teóricos e

práticos, melhorias podem ser aplicadas.

Em todos os testes apresentados, realizou-se um particionamento único em que, dada a representação do grafo de conflitos, este era particionado uma única vez. Dependendo da natureza do problema, pode ser mais interessante o particionamento hierárquico, ou seja, particiona-se o grafo de conflitos em dois, em seguida cada partição é novamente particionada em dois, e assim sucessivamente até obter o número desejado de partições. Essa estratégia pode permitir com que a heurística de particionamento seja mais eficiente, reduzindo o tamanho do corte.

Outra técnica interessante que pode ser utilizada na LagClus, é a clonagem de vértices apresentada por Sachdeva (2004). A idéia consiste em, dada a divisão do grafo de conflitos em *clusters*, os vértices correspondentes às arestas de conexão são copiados (clonados) de tal maneira que essas arestas apareçam nos subproblemas, reduzindo o número de arestas relaxadas. Essa estratégia pode ser útil também para o AGC. Atualmente encontra-se em desenvolvimento a aplicação desta técnica na LagClus para o problema do PCPP.

No caso do AGC, técnicas de eliminação de colunas improdutivas devem ser incorporadas, pois em alguns casos, este método torna-se lento. O uso de heurísticas mais elaboradas pode ser uma estratégia interessante, e, conseqüentemente, encontra-se em estudo novas heurísticas para os problemas aqui estudados.

Como o AGC apresentou bons resultados, pretende-se elaborar um algoritmo de *Branch-and-Price*. As estratégias existentes de separação e descida na árvore de busca serão estudadas, objetivando realizar a maior parte das podas na parte superior da árvore. Também serão estudadas as técnicas existentes para a geração de colunas em cada nó da árvore. Espera-se obter um método eficiente capaz de resolver problemas de larga escala.

Considerando o PRCP, uma nova modelagem matemática para este problema que considera o PMNRSC, está sendo desenvolvida. Estudos iniciais com o CPLEX nas instâncias de 1000 pontos mostraram que é possível aumentar o número de rótulos livres em mais de 1%, mesmo não resolvendo de forma ótima nenhuma das 25 instâncias. Conseqüentemente, cortes estão sendo estudados de tal maneira a fortalecer a formulação encontrada. Dependendo do número de novas restrições encontradas, pretende-se fazer uma aplicação do Algoritmo de Planos de Corte.

REFERÊNCIAS BIBLIOGRÁFICAS

- Abou-Rjeili, A.; Karypis, G. Multilevel algorithms for partitioning power-law graphs. In: International Parallel and Distributed Processing Symposium, IPDPS 2006, 20., 2006. **Proceedings...** Rhodes Island, Grécia: IEEE, 2006. [54](#), [55](#)
- Alvarez-Valdes, R.; Parreño, F.; Tamarit, J. M. A branch-and-cut algorithm for the pallet loading problem. **Computers and Operations Reserach**, v. 32, n. 11, p. 3007–3029, 2005a. [22](#), [48](#), [112](#), [115](#), [125](#), [126](#), [129](#)
- . A tabu search algorithm for pallet loading problem. **OR Spectrum**, v. 27, n. 1, p. 43–61, 2005b. [112](#)
- Alvim, A. C. F.; Taillard, E. D. POPMUSIC for the point feature label placement. In: Metaheuristics International Conference, MIC 2005, 6., 2005. **Proceedings...** Viena, Aústria, 2005. p. 39–44. [80](#), [81](#), [82](#), [86](#), [108](#)
- Atamtürk, A.; Nemhauser, G. L.; Savelsbergh, M. W. P. Conflict graphs in solving integer programming problems. **European Journal of Operational Research**, v. 121, p. 40–55, 2000. [47](#)
- Balas, E.; Padberg, M. Set partitioning. **SIAM Review**, v. 18, p. 710–760, 1976. [51](#)
- Barnes, F. W. Packing the maximum number of $m \times n$ tiles in a large $p \times q$ rectangles. **Discrete Mathematics**, v. 26, p. 93–100, 1979. [115](#), [127](#)
- Barnhart, C.; Johnson, E.; Nemhauser, G.; Savelsbergh, M.; Vance, P. Branch-and-price: column generation for solving huge integer programs. **Operations Research**, v. 46, p. 316–329, 1998. [40](#), [42](#)
- Beasley, J. An exact two-dimensional non guillotine cutting tree search procedure. **Operations Research**, v. 33, p. 49–64, 1985. [18](#), [115](#), [117](#), [119](#), [120](#), [122](#), [124](#), [134](#), [174](#)
- Benoist, T.; Rottembourg, B. Upper bounds for revenue maximization in a satellite scheduling problem. **Quarterly Journal of the Belgian, French and Italian Operations Research Societies**, v. 2, n. 3, p. 235–249, 2004. [141](#)
- Bensana, E.; Lemaitre, M.; Verfaillie, G. Benchmark problems: earth observation satellite management. **Constraints**, v. 4, n. 3, p. 293–299, 1999. [31](#), [141](#), [142](#), [143](#), [163](#), [172](#)

Bensana, E.; Verfaillie, G.; Agnèsse, J. C.; Bataille, N. Blumstein, D. Exact and approximate methods for the daily management of an earth observation satellite. In: International Symposium on Space Mission Operations and Ground Data Systems, SpaceOps-96, 4., 1996. **Proceedings...** Munique, Alemanha, 1996. Disponível em: ftp://ftp.cert.fr/pub/verfaillie/spaceops96.ps. Acesso em: 14 nov. 2005. [142](#), [143](#)

Bhattacharya, R.; Roy, R.; Bhattacharya, S. An exact depth-first algorithm for the pallet loading problem. **European Journal of Operational Research**, v. 110, p. 610–625, 1998. [112](#)

Birgin, E. G.; Morabito, R.; Nishihara, F. H. A note on an L-approach for solving the manufacturer's pallet loading problem. **Operations Research Letters**, v. 56, p. 1448–1451, 2005. [112](#)

Bischoff, E.; Dowsland, W. B. An application of the micro to product design and distribution. **Journal of the Operational Research Society**, v. 33, n. 3, p. 271–280, 1982. [113](#), [114](#)

Björkund, P.; Värbrand, P.; Yuan, D. Optimized planning of frequency hopping in cellular networks. **Computers and Operations Reserach**, v. 32, p. 169–186, 2005. [48](#)

Branch, A. E. **Elements of shipping**. London: Chapman & Hall, 1996. 240 p. [132](#)

Briant, O.; Lemaréchal, C.; Meurdesoif, P.; Michel, S.; Perrot, N.; Vanderbeck, F. **Comparison of bundle and classical column generation**. Rhone-Alpes, França: INRIA, 2005. Rapport de recherche de l'INRIA 5453. [73](#)

Bui, T. N.; Moon, B. R. Genetic algorithm and graph partitioning. **IEEE Transactions on Computers**, v. 45, n. 7, p. 841–855, 1996. [53](#)

Carvalho, J. M. C. Using extra dual cuts to accelerate convergence in column generation. **INFORMS Journal on Computing**, v. 17, n. 2, p. 175–182, 2005. [42](#)

Cheng, E.; Cunningham, W. H. Wheel inequalities for stable set polytopes. **Mathematical Programming**, v. 77, p. 389–421, 1997. [17](#), [50](#), [51](#)

Christensen, J.; Marks, J.; Shieber, S. **Placing text labels on maps and diagrams**. London: Academic Press, 1993. [80](#)

———. An empirical study of algorithms for point-features label placement. **ACM Transactions on Graphics**, v. 14, n. 3, p. 203–232, 1995. [17](#), [78](#), [80](#), [108](#)

- Christofides, N.; Whitlock, C. An Algorithm for Two-Dimensional Cutting Problems. **Operations Research**, v. 25, n. 1, p. 30–44, 1977. [116](#)
- Churh, R. L.; Murray, A. T.; Weintraub, A. Locational issues in forest management. **Location Science**, v. 6, p. 137–153, 1998. [48](#)
- Cánovas, L.; Landete, M.; Marín, A. New facets for the set packing polytope. **Operations Research Letters**, v. 27, n. 4, p. 153–161, 2000. [51](#)
- Constantino, M. F.; Martins, I.; Borges, J. A new mixed integer programming model for harvest scheduling subject to maximum area restrictions. **Operations Research**, 2007. To appear. [48](#)
- Cooley, R.; Mobasher, B.; Srivastava, J. Web mining: information and pattern discovery on the world wide web. In: IEEE International Conference on Tools with Artificial Intelligence, ICTAI97, 1997. **Proceedings...** Newport Beach, CA, 1997. [52](#)
- Cordeau, J.-F.; Laporte, G. Maximizing the value of an earth observation satellite orbit. **Journal of the Operational Research Society**, v. 56, p. 962–968, 2005. [141](#)
- Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C. **Algoritmos: teoria e prática**. Campus, 2002. 936 p. [59](#)
- Cornuejols, G.; Thizy, J.-M. Some facets of the simple plant location polytope. **Mathematical Programming**, v. 23, p. 50–74, 1982. [50](#)
- 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**, 2007. To appear. [81](#), [86](#), [108](#)
- Dantzig, G. B.; Wolfe, P. Decomposition principle for linear programs. **Operations Research**, v. 8, p. 101–111, 1960. [39](#)
- Desrosiers, J.; Lübbecke, M. E. A primer in column generation. In: Desaulniers, G.; Desrosiers, J.; Solomon, M. M. (eds.) **Column generation**. New York: Springer-Verlag, 2005. cap. 1, p. 1–32. [42](#), [105](#)
- Diestel, R. **Graph theory**. Springer-Verlag, 2000. 436 p. [59](#)
- Dowland, K. A. A combined data-base and algorithmic approach to the pallet-loading. **Journal of the Operations Research Society**, v. 38, n. 4, p. 341–345, 1987a. [113](#)

- . An exact algorithm for the pallet loading problem. **European Journal of Operational Research**, v. 31, p. 78–84, 1987b. [22](#), [31](#), [48](#), [112](#), [115](#), [125](#), [126](#), [128](#)
- Dyckhoff, H. A typology of cutting and packing problems. **European Journal of Operational Research**, v. 31, p. 78–84, 1990. [111](#), [174](#)
- Ferreira, C.; Wakabayashi, Y. **Combinatória poliédrica e planos-de-corte faciais**. Unicamp, SP: Unicamp, 1996. 130 p. [45](#)
- Fjällström, P. O. **Algorithms for graph partitioning: a survey**. Computer and Information Science: Linköping University, 1998. Vol. 3, Nr. 10. Disponível em: <http://www.ep.liu.se/ea/cis/1998/010>. Acesso em: 22 jan. 2004. [52](#), [53](#)
- Formann, M.; Wagner, F. A packing problem with applications to lettering of maps. In: International ACM Symposium on Computational Geometry, 7., 1991. **Proceedings...** New Hampshire, EUA: ACM, 1991. p. 281–288. [77](#)
- Fréville, A.; Hanafi, S. The multidimensional 0-1 knapsack problem - bounds and computational aspects. **Annals of Operations Research**, v. 139, n. 1, p. 195–227, 2005. [145](#)
- Fulkerson, D. Blocking and anti-blocking pairs of polyhedra. **Mathematical Programming**, v. 1, p. 168–194, 1971. [154](#)
- Gabrel, V. **Improved linear programming bounds via column generation procedure for the daily scheduling of earth observation satellite**. LIPN: Paris XIII University, 1999. Research Report 99-01. [171](#)
- . An extensive comparison of 0-1 linear programs for SPOT5 daily mission planning problem. **Annales du LAMSADE**, v. 4–5, p. 283–293, 2005. [142](#), [171](#)
- . Strengthened 0-1 linear formulation for the daily satellite mission planning. **Journal of Combinatorial Optimization**, v. 11, p. 351–346, 2006. [142](#)
- Gabrel, V.; Murat, C. Mathematical programming for earth observation satellite mission planning. In: Ciriani, T.; Fasano, G.; Gliozzi, S.; Tadei, R. (eds.) **Operations research in space and air**. Dordrecht: Kluwer Academic Publishers, 2003. p. 103–122. [141](#), [142](#)
- Garey, M.; Johnson, D.; Stockmeyer, L. Some simplified NP-complete graph problems. **Theoretical Computer Science**, v. 1, p. 237–267, 1976. [52](#)
- Gerrard, R.; Church, R. L. Closest assignment constraints and location models: properties and structures. **Location Science**, v. 4, p. 251–270, 1996. [48](#)

- Gilbert, J. R.; Zmijewski, E. A parallel graph partitioning algorithm for a message-passing multiprocessor. **International Journal of Parallel Programming**, v. 16, n. 1, p. 427–449, 1987. [54](#)
- Glover, F. Surrogate constraints. **Operations Research**, v. 16, p. 741–749, 1968. [87](#)
- Goldberg, M. C.; Luna, H. P. L. **Otimização combinatória e programação linear**. Rio de Janeiro, RJ: Campus, 2005. 536 p. [92](#)
- Gomory, R. Outline of an algorithm for integer solutions to linear programs. **Bulletin of the American Mathematical Society**, v. 64, p. 275–278, 1958. [42](#)
- Goycoolea, M.; Murray, A. T.; Barahona, F.; Epstein, R.; Weintraub, A. Harvest scheduling subject to maximum area restrictions: exploring exact approaches. **Operations Research**, v. 53, n. 3, p. 490–500, 2005. [48](#)
- Guignard, M. Lagrangean relaxation. **TOP**, v. 11, n. 2, p. 151–228, 2003. [33](#), [37](#)
- Habet, D.; Vasquez, M. Solving the selecting and scheduling satellite photographs problem with a consistent neighborhood heuristic. In: International Conference on Tools with Artificial Intelligence, ICTAI 2004, 16., 2004. **Proceedings...** Boca Raton, EUA: IEEE, 2004. p. 302–309. [141](#)
- Handabaka, A. R. **Gestão logística da distribuição física internacional**. São Paulo: Maltese, 1994. 450 p. [132](#)
- Harary, F. **Graph theory**. New York: Addison-Wesley, 1972. [48](#), [49](#)
- Held, M.; Karp, R. M. The traveling salesman problem and minimum spanning trees. **Operations Research**, v. 18, p. 1138–1162, 1970. [35](#)
- . The traveling salesman problem and minimum spanning trees: part II. **Mathematical Programming**, v. 1, p. 6–25, 1971. [35](#), [98](#)
- Herbert, A.; Dowsland, W. A family of genetic algorithms for the pallet loading problem. **Annals of Operations Research**, v. 63, p. 415–436, 1996. [112](#)
- Hicks, I. V.; Warren, J. S.; Warriar, D.; Wilhenlm, W. E. **A Branch-and-Price approach for the maximum weight independent set problem**. Texas A & M University: Department of Industrial Engineering, 2004. Disponível em: <http://ie.tamu.edu/People/faculty/Hicks/>. Acesso em: 22 jan. 2004. [61](#)
- Hirsh, S. A. An algorithm for automatic name placement around point data. **American Cartographer**, v. 9, n. 1, p. 5–17, 1982. [31](#), [108](#)

- Hodgson, T. A combined approach to the pallet loading problem. **IIE Transactions**, v. 14, n. 3, p. 176–182, 1982. [31](#), [111](#)
- CPLEX Reference manual. França, 2001a. 610 p. [45](#), [86](#)
- CPLEX Reference manual. France, 2006b. [103](#)
- Isermann, H. Ein Planungssystem zur Optimierung der Palettenbeladung mit kongruenten rechteckigen Versandgebinden. **OR Spectrum**, v. 9, p. 235–249, 1987. [115](#)
- Johnson, D. S.; Aragon, C. R.; McGeoch, L. A.; Schevon, C. Optimization by simulated annealing: an experimental evaluation. Part I, graph partitioning. **Operations Research**, v. 37, n. 6, p. 865–892, 1989. [53](#)
- Karisch, S. E.; Rendl, F.; Clausen, J. **Solving graph bi-section problems with semidefinite programming**. Copenhagen, Department of Computer Science: University of Copenhagen, 1997. DIKU-TR-97/9. [52](#)
- Karypis, G. **Multilevel hypergraph partitioning**. Minnesota, Department of Computer Science: University of Minnesota, 2002. TR 02–25. Disponível em: <http://www-users.cs.umn.edu/~karypis/publications/partitioning.html>. Acesso em: 30 mar. 2005. [17](#), [54](#), [55](#)
- . Multi-Constraint Mesh Partitioning for Contact/Impact Computations. In: ACM/IEEE International Conference on Supercomputing, 2003. **Proceedings...** Phoenix, EUA: IEEE, 2003. 56 CD-ROM. [56](#)
- Karypis, G.; Kumar, V. A fast and high quality multilevel scheme for partitioning irregular graphs. **SIAM Journal on Scientific Computing**, v. 20, p. 359–392, 1998a. [53](#), [54](#), [55](#), [68](#)
- . **Multilevel algorithms for multi-constraint graph partitioning**. Minnesota, Army HPC Research Center, Dep. of Computer Science: University of Minnesota, 1998b. Technical Report 98-019. [53](#), [54](#), [55](#), [101](#)
- . Multilevel k-way partitioning scheme for irregular graphs. **Journal of Parallel and Distributed Computing**, v. 48, n. 1, p. 96–129, 1998c. [52](#), [55](#)
- . Multilevel k-way hypergraph partitioning. In: Design and Automation Conference, 36., 1999. **Proceedings...** IEEE, 1999. [52](#)
- Kernighan, B. W.; Lin, S. An efficient heuristic procedure for partitioning graphs. **The Bell System Technical J.**, v. 49, p. 291–307, 1970. [27](#), [53](#)

- Lemaître, M.; Verfaillie, G. Daily management of an earth observation satellite: comparison of ILOG solver with dedicated algorithms for valued constraint satisfaction problems. In: ILOG International Users Meeting, 3., 1997. **Proceedings...** Paris, França: ILOG, 1997. Disponível em: ftp://ftp.cert.fr/pub/verfaillie/ilog97.ps. Acesso em: 20 nov. 2006. [142](#)
- Letchford, A. N.; Amaral, A. Analysis of upper bounds for the pallet loading problem. **European Journal of Operational Research**, v. 3, n. 132, p. 582–593, 2001. [22](#), [114](#), [125](#), [126](#), [128](#), [138](#)
- Lin, W.-C.; Liao, D.-Y.; Liu, C.-Y.; Lee, Y.-Y. Daily imaging scheduling of an earth observation satellite. **IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans**, v. 35, p. 213–223, 2005. [141](#)
- Lins, L.; Lins, S.; Morabito, R. A L-approach for packing (l-w)rectangles into rectangular and L-shaped pieces. **Journal of the Operational Research Society**, v. 54, p. 777–789, 2003. [112](#)
- Lorena, L.; Pereira, M.; Salomão, S. A relaxacao Lagrangeana/surrogate e o metodo de geracao de colunas: novos limitantes e novas colunas. **Pesquisa Operacional**, v. 23, n. 1, p. 29–47, 2003. [73](#)
- Lorena, L. A. N.; Furtado, J. C. Constructive genetic algorithm for clustering problems. **Evolutionary Computation**, v. 3, n. 9, p. 309–327, 2001. [80](#)
- Marks, J.; Shieber, S. **The computational complexity of cartographic label placement**. Harvard University: Center for Research in Computing Technology, 1991. [77](#)
- Marsten, R. M.; Hogan, W.; Blankenship, J. The boxstep method for large-scale optimization. **Operations Research**, v. 23, p. 389–405, 1975. [73](#)
- Martello, S.; Toth, P. **Knapsack problems: algorithms and computer implementations**. John Wiley & Sons Inc, 1990. 308 p. [145](#)
- Martins, G. H. **A Packing in two and three dimensions**. 2003. 176 p. Tese (Doutorado em Operations Reserach) – Naval Postgraduate School, Monterey, CA - EUA. 2003. [18](#), [113](#)
- du Merle, O.; Goffin, J.; Vial, J. On improvements to the analytic centre cutting plane method. **Computational Optimization and Applications**, v. 11, p. 37–52, 1998. [73](#)

- Miller, G. L.; Teng, S.-H.; Thurston, W.; Vavasis, S. A. Graph theory and sparse matrix computation. In: George, A.; Gilbert, J. R.; Liu, L. W. H. (eds.) **The IMA volumes in mathematics and its applications**. Springer Verlag, 1993. p. 57–84. [53](#)
- Moon, I. D.; Chaudhry, S. An analysis of network location problems with distance constraints. **Management Science**, , n. 30, p. 290–307, 1984. [80](#)
- Morabito, R.; Farago, R. A tight lagrangean relaxation bound for the manufacturer's pallet loading problem. **Studia Informatica Universalis**, v. 2, n. 1, p. 57–76, 2002. [115](#), [126](#)
- Morabito, R.; Morales, S. R. A simple and effective heuristic to solve the manufacturer's pallet loading problem. **Journal of the Operational Research Society**, v. 49, p. 819–828, 1998. [18](#), [112](#), [115](#), [125](#)
- Morabito, R.; Morales, S. R.; Widmer, J. A. Loading optimization of palletized products on trucks. **Transportation Research Part E**, v. 36, p. 285–296, 2000. [126](#)
- Murray, A. T.; Church, R. L. Constructing and selecting adjacency constraints. **INFOR**, p. 232–248, 1996. [84](#)
- . Facets for node packing. **European Journal of Operational Research**, v. 101, p. 598–608, 1997a. [19](#), [48](#), [149](#), [150](#), [152](#), [153](#)
- . Solving the anti-covering location problem using lagrangean relaxation. **Computers and Operations Research**, v. 24, n. 2, p. 127–140, 1997b. [80](#), [81](#)
- Narciso, M. G.; Lorena, L. A. N. Lagrangean/Surrogate relaxation for generalized assignment problems. **European Journal of Operational Research**, v. 114, p. 165–177, 1999. [29](#), [73](#), [87](#), [98](#), [126](#), [163](#)
- Neame, P. **Nonsmooth dual methods in integer programming**. 1999. 172 p. Tese (Doutorado em Mathematics and Statistics) – University of Melbourne, Melbourne, Austrália. 1999. [73](#)
- Nemhauser, G.; Sigismondi, G. A strong cutting plane/branch-and-bound algorithm for node packing. **Journal of Operations Research**, v. 43, p. 443–457, 1992. [31](#), [49](#), [153](#)
- Oliveira, L. K. **Métodos exatos baseados em relaxações lagrangiana e surrogate para o problema de carregamento de paletes do produtor**. 2005.

- Tese (Doutorado em Engenharia de Produção) – Universidade de São Paulo, São Carlos. 2005. [117](#)
- Padberg, M. W. On the facial structure of set packing polyhedra. **Mathematical Programming**, v. 5, p. 199–215, 1973. [50](#), [154](#)
- Parker, R. G.; Rardin, R. L. **Discrete optimization**. New York: Academic Press, 1988. [17](#), [29](#), [33](#), [34](#), [37](#), [38](#), [45](#)
- Pereira, M. A. **Um método branch-and-price para problemas de localização de p-medianas**. 2005. 90 p. Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos-SP, Brasil. 2005. [73](#)
- Pereira, M. A.; Lorena, L. A. N.; Senne, E. L. F. A column generation approach for the maximal covering location problem. **International Transactions in Operations Research**, v. 14, p. 349–364, 2007. [42](#)
- Preiss, B. **Data structures and algorithms with object-oriented design patterns in C++**. Waterloo, University of Waterloo, Canada: John Wiley & Son, 1998. 688 p. [59](#)
- Pureza, V.; Morabito, R. Some experiments with a simple tabu search algorithm for the manufacturer’s pallet loading problem. **Computers and Operations Research**, v. 33, n. 3, p. 804–819, 2006. [112](#), [125](#)
- Ribeiro, G. M.; Constantino, M. F.; Lorena, L. A. N. New formulations and optimal solutions for the spot 5 daily photograph scheduling problem. **European Journal of Operational Research**, 2007. Submitted. [175](#)
- Ribeiro, G. M.; Lorena, L. A. N. Modelagem matemática e relaxações Lagrangeana e Lagrangeana/Surrogate para o problema de rotulação cartográfica de pontos. In: Simpósio Brasileiro de Pesquisa Operacional (SBPO), 36., 2004, São João del Rei. **Anais...** Rio de Janeiro: SOBRAPO, 2004a. 1 CD-ROM. Disponível em: <http://www.lac.inpe.br/~lorena/glaydston/Glaydston-lorena-Sbpo.PDF>. Acesso em: 20 jan. 2005. [78](#), [81](#), [174](#)
- . Uma nova modelagem matemática para o problema da rotulação cartográfica de pontos. In: IV WORCAP, 2004, São José dos Campos. **Proceedings...** São Paulo: INPE, 2004b. [78](#), [81](#)
- . Heuristics for cartographic label placement problems. **Computers & Geoscience**, v. 32, n. 6, p. 739–748, 2006a. [174](#), [175](#)

- . Lagrangean relaxation bounds for point-feature cartographic label placement problem. **Pesquisa Operacional**, v. 26, n. 3, p. 459–471, 2006b. [62](#)
- . Lagrangean relaxation with clusters for point-feature cartographic label placement problems. **Computers and Operations Research**, 2006c. To appear. [62](#), [174](#), [175](#)
- . Optimizing the woodpulp stowage using lagrangean relaxation with clusters. **Journal of the Operational Research Society**, 2006d. To appear. [31](#), [62](#), [174](#), [175](#)
- . Column generation approach for the point-feature cartographic label placement problem. **Journal of Combinatorial Optimization**, 2007a. To Appear. [176](#)
- . Lagrangean relaxation with clusters and column generation for the manufacturer’s pallet loading problem. **Computers and Operations Research**, v. 34, n. 9, p. 2695–2708, 2007b. [62](#), [175](#), [176](#)
- Rolland, E.; Pirkul, H.; Glover, F. Tabu search for graph partitioning. **Annals of Operations Research**, v. 63, p. 209–232, 1996. [53](#)
- Sachdeva, S. **Development of a branch and price approach involving vertex cloning to solve the maximum weighted independent set problem**. 2004. Dissertação (Mestrado em Industrial Engineering) – Texas A&M University, Texas, EUA, 2004. [177](#)
- Scheithauer, G.; Terno, J. The G4-heuristic for the pallet loading problem. **Journal of the Operational Research Society**, v. 47, p. 511–522, 1996. [112](#)
- Schloegel, K.; Karypis, G.; Kumar, V. **Multilevel diffusion schemes for repartitioning of adaptive meshes**. Minnesota, Department of Computer Science: University of Minnesota, 1997. TR–97–013. [54](#)
- Schreyer, M.; Raidl, G. R. Letting ants labeling point features. In: IEEE Congress on Evolutionary Computation at the IEEE World Congress on Computational Intelligence, 2004. **Proceedings...** IEEE, 2004. p. 1564–1569. [80](#)
- Senne, E. L. F.; Lorena, L. A. N. Lagrangean/Surrogate heuristics for p-median problems. In: Laguna, M.; Gonzalez-Velarde, J. L. (eds.) **Computing tools for modeling, optimization and simulation: interfaces in computer science and operations research**. Netherlands: Kluwer Academic Publisher, 2000. p. 115–130. [88](#), [99](#)

- Senne, E. L. F.; Lorena, L. A. N.; Pereira, M. A. A simple stabilizing method for column generation heuristics: an application to p-median location problems. **International Journal of Operations Research**, v. 4, p. 1–9, 2007. [42](#), [72](#), [73](#)
- Senne, L. F.; Lorena, L. A. N.; Pereira, M. A. A branch-and-price approach to p-median location problems. **Computers and Operations Research**, v. 32, p. 1655–1664, 2005. [72](#), [73](#)
- Shapiro, J. A survey of lagrangean techniques for discrete optimization. **Annals of Discrete Mathematics**, v. 5, p. 113–138, 1974. [37](#)
- Shekhar, S.; Liu, D. R. Partitioning similarity graphs: A framework for declustering problems. **Information Systems Journal**, v. 21, n. 6, p. 475–496, 1996. [52](#)
- Silva, J. L. C.; Soma, N. Y.; Maculan, N. A greedy search for the three-dimensional bin packing problem: the packing static stability case. **International Transactions in Operational Research**, v. 10, p. 1–13, 2003. [133](#)
- Simon, H. D. Partitioning of unstructured problems for parallel processing. **Computing Systems in Engineering**, v. 2, n. 2–3, p. 135–148, 1991. [53](#)
- Smith, A.; De Cani, P. An algorithm to optimize the layout of boxes in pallets. **Journal of the Operational Research Society**, v. 31, p. 573–578, 1980. [112](#), [114](#), [122](#)
- Studel, H. J. Generating pallet loading patterns: a special case of the two-dimensional cutting stock problem. **Management Science**, v. 25, p. 997–1004, 1979. [112](#)
- Strijk, T.; Verweij, B.; Aardal, K. **Algorithms for maximum independent set applied to map labeling**. Universiteit Utrecht: Institute of Information and Computing Sciences, 2000. Disponível em: [<ftp://ftp.cs.uu.nl/pub/ruu/cstechreps/cs-2000/2000-22.ps.gz>](ftp://ftp.cs.uu.nl/pub/ruu/cstechreps/cs-2000/2000-22.ps.gz). Acesso em: 10 nov. 2003. [79](#), [80](#)
- Taillard, E.; Voss, S. POPMUSIC: Partial optimization metaheuristic under special intensification conditions. In: Ribeiro, C.; Hansen, P. (eds.) **Essays and Surveys in Metaheuristics**. Boston: Kluwer Academic Publishers, 2001. p. 613–629. [80](#)
- Torres, F. P. **Algoritmos heurísticos y exactos para problemas de corte no guillotina en dos dimensiones**. 2004. Tese (Doutorado em Ciências Matemáticas) – Universitat de València, València, Espanha. 2004. [51](#)

- Vasquez, M.; Hao, J. K. A logic-constrained knapsack formulation and a tabu algorithm for daily photograph scheduling of an earth observation satellite. **Journal of Computational Optimization and Applications**, v. 20, p. 137–157, 2001. [141](#), [143](#), [144](#), [145](#), [146](#), [148](#), [151](#), [153](#), [154](#), [160](#), [163](#), [166](#), [167](#), [172](#)
- . Upper bounds for the SPOT 5 daily photograph scheduling problem. **Journal of Combinatorial Optimization**, v. 7, p. 87–103, 2003. [143](#), [145](#), [146](#), [148](#), [165](#), [171](#), [172](#)
- Verfaillie, G.; Lemaitre, M.; Schiex, T. Russian doll search for solving constraint optimization problems. In: International Conference on Artificial Intelligence, 13., 1996. **Proceedings...** Portland, EUA, 1996. p. 182–187. [142](#)
- Verner, O. V.; Wainwright, R. L.; Schoenefeld, D. A. Placing text labels on maps and diagrams using genetic algorithms with masking. **INFORMS Journal on Computing**, v. 9, p. 266–275, 1997. [80](#), [96](#), [108](#)
- Wagner, F.; Wolff, A.; Kapoor, V.; Strijk, T. Three rules suffice for good label placement. **Algorithmica**, v. 30, p. 334–349, 2001. [18](#), [82](#), [96](#), [103](#), [109](#)
- Walshaw, C. Multilevel refinement for combinatorial optimisation problems. **Annals of Operations Research**, v. 131, p. 325–372, 2004. [54](#)
- Walshaw, C.; Cross, M.; Everett, M. G. Parallel dynamic graph partitioning for adaptive unstructured meshes. **Journal of Parallel and Distributed Computing**, v. 47, n. 2, p. 102–108, 1997. Disponível em: <http://www.gre.ac.uk/~c.walshaw/jostle/>. [54](#)
- Wolff, A. **Automated label placement in theory and practice**. 1999. Tese (Doutorado em Fachbereich Mathematik und Informatik) – Freie Universität Berlin, Berlin. 1999. [77](#)
- Wolff, A.; Strijk, T. The map labeling bibliography, 2006. Disponível em: <http://i11www.ilkd.uni-karlsruhe.de/~awolff/map-labeling/bibliography/>. Acesso em: 22 jan. 2004. [77](#)
- Wolsey, L. A. **Integer programming**. New York: Jhon Wiley & Sons, 1998. [29](#), [33](#), [34](#), [37](#), [41](#), [42](#), [44](#), [45](#), [46](#)
- XPRESS Reference manual. Reino Unido, 2006. [46](#)

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. [80](#), [81](#), [82](#), [86](#), [92](#), [96](#), [108](#)

Yamamoto, M.; Lorena, L. A. N. A constructive genetic approach to point-feature cartographic label placement. In: Ibaraki, T.; Nonobe, K.; Yagiura, M. (eds.) **Metaheuristics: progress as real problem solvers**. Netherlands: Kluwer Academic Publishers, 2005. p. 285–300. [48](#), [80](#), [81](#), [82](#), [86](#), [96](#), [108](#), [124](#)

Young-Gun, G.; Maing-Kyu, K. A fast algorithm for two-dimensional pallet loading problems of large size. **European Journal of Operational Research**, v. 134, p. 193–202, 2001. [112](#)

Zoraster, S. Integer programming applied to the map label placement problem. **Cartographica**, v. 23, n. 3, p. 16–27, 1986. [79](#)

———. The solution of large 0-1 integer programming problems encountered in automated cartography. **Operations Research**, v. 38, n. 5, p. 752–759, 1990. [31](#), [48](#), [79](#), [80](#), [84](#), [86](#), [96](#), [108](#)

———. Expert systems and the map label placement problem. **Cartographica**, v. 28, n. 1, p. 1–9, 1991. [79](#), [80](#)

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programa de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. São aceitos tanto programas fonte quanto executáveis.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.