



**Ministério da
Ciência e Tecnologia**



INPE-15481-TDI/1424

**NOVAS ABORDAGENS PARA REPRESENTAÇÃO E
OBTENÇÃO DE LIMITANTES E SOLUÇÕES PARA
ALGUNS PROBLEMAS DE OTIMIZAÇÃO
COMBINATÓRIA**

Geraldo Regis Mauri

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada,
orientada pelo Dr. Luiz Antonio Nogueira Lorena, aprovada em 28 de novembro de
2008

Registro do documento original:
<<http://urlib.net/sid.inpc.br/mte-m18@80/2008/11.02.01.55>>

INPE
São José dos Campos
2008

Dados Internacionais de Catalogação na Publicação (CIP)

M446n Mauri, Geraldo Regis.

Novas abordagens para representação e obtenção de limitantes e soluções para alguns problemas de otimização combinatória / Geraldo Regis Mauri. – São José dos Campos: INPE, 2008. 239p. ; (INPE-15481-TDI/1424)

Tese (Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2008.

1. Dial-a-ride. 2. Alocação de berços. 3. Programação quadrática. 4. Rotulação cartográfica. 5. Geração de colunas. 6. Relaxações. 7. Programa linear. I.Título.

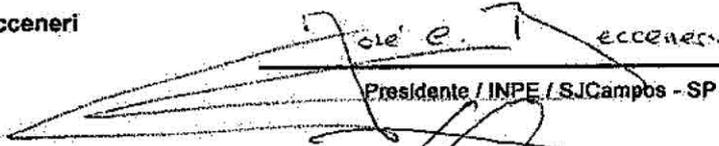
CDU 519.853

Copyright © 2008 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, microfílmico, reprográfico ou outros, sem a permissão escrita da Editora, com exceção de qualquer material fornecido especificamente no propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2008 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, microfilming, recording or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

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

Dr. José Carlos Becceneri



Jose C. Becceneri

Presidente / INPE / SJC Campos - SP

Dr. Luiz Antonio Nogueira Lorena



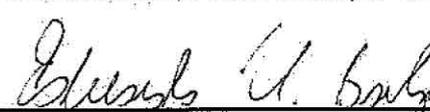
Orientador(a) / INPE / SJC Campos - SP

Dr. Stephan Stephany



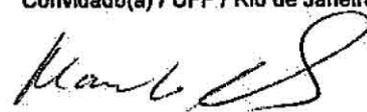
Membro da Banca / INPE / SJC Campos - SP

Dr. Eduardo Uchoa Barboza



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

Dr. Marcos Nereu Arenales



Convidado(a) / USP-São Carlos / São Carlos - SP



Aluno (a): Geraldo Regis Mauri

São José dos Campos, 28 de novembro de 2008

“O único lugar onde sucesso vem antes de trabalho é no dicionário”.

ALBERT EINSTEIN

A meus pais e minha esposa...

AGRADECIMENTOS

A Deus, pela saúde, proteção e oportunidade de chegar até aqui, estando sempre ao meu lado.

A meus pais, por sempre acreditarem na importância do estudo.

À minha família, em especial à minha avó Tereza, pelo apoio e incentivo incondicional durante todas as etapas da vida.

À minha esposa Larice, pelo amor e carinho, pela compreensão e apoio, e principalmente por estar sempre ao meu lado nos melhores e piores momentos.

Ao orientador professor Luiz Antonio Nogueira Lorena, pela diversidade de conhecimentos transmitidos e pela orientação e apoio contínuo à realização deste trabalho.

Ao professor Marcone Jamilson Freitas Souza (UFOP), pela empolgação e incentivo à realização do mestrado e doutorado.

Aos professores Glaydston Mattos Ribeiro (UFES) e Alexandre César Muniz de Oliveira (UFMA) pelas valiosas contribuições em partes desta pesquisa.

Ao Instituto Nacional de Pesquisas Espaciais (INPE) pela oportunidade, e aos professores pelos ensinamentos transmitidos, em especial aos professores Horacio Hideki Yanasse, Stephan Stephany e Fernando Manuel Ramos.

Aos colegas do Centro de Ciências Agrárias (CCA) da Universidade Federal do Espírito Santo (UFES) pela receptividade.

À grandiosa e eterna República *VIRA SAIA* (Ouro Preto) pela minha formação, aprendizado e convivência, e aos irmãos “Canalhas”, pela amizade, companheirismo e conhecimentos transmitidos.

Aos grandes amigos Glaydston, Caê, Antônio, Corrêa, Henrique, Sidão, Thiago, Gilberto e Reinaldo, pelo convívio e troca de experiências ao longo desta jornada.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pelo auxílio financeiro que viabilizou o desenvolvimento deste trabalho.

RESUMO

Esta tese apresenta novas estratégias para representar e obter soluções para quatro diferentes problemas de Otimização Combinatória (OC). Além disso, são apresentadas também algumas alternativas para verificar a qualidade de soluções já conhecidas. Inicialmente, é proposta uma nova forma para representação de um problema de roteamento de veículos conhecido como *Dial-a-Ride* (DARP), e a partir dessa representação é aplicado um método baseado na metaheurística *Simulated Annealing* (SA) para obter soluções. A representação proposta oferece uma certa flexibilidade na escolha do objetivo final a ser otimizado por meio de uma função multi-objetivo, o que permite tratar diversas especificidades do problema. O método proposto apresentou excelentes resultados, e comparado à literatura obteve, em todos os casos, melhores soluções sob o ponto de vista prático do problema. Na sequência, a mesma abordagem proposta para o DARP é adaptada e aplicada ao Problema de Alocação de Berços (PAB), comum em sistemas portuários. O Algoritmo de Treinamento Populacional (ATP) e sua integração com a Programação Linear (PL) por meio da técnica de Geração de Colunas (ATP/PL) também são aplicados ao PAB. A qualidade das soluções obtidas é verificada com a utilização de um *software* comercial para o qual estas são utilizadas como soluções iniciais. Além disso, os resultados ainda são comparados com uma abordagem recente encontrada na literatura, apresentando melhores soluções. Diversas abordagens visando à obtenção de limitantes para o Problema de Programação Quadrática Binária Irrestrita (PQ) também são apresentadas. Novas alternativas de Relaxação Lagrangiana com *Clusters* e de Decomposição Lagrangiana são utilizadas para tratar o PQ. A Decomposição Dantzig-Wolfe também é aplicada em conjunto com esses métodos para obtenção de soluções e limitantes. Os resultados obtidos demonstram que os limitantes duais obtidos são melhores que os apresentados por relaxações lagrangianas tradicionais, e as soluções obtidas melhores do que as apresentadas em alguns trabalhos encontrados na literatura. Por fim, a Decomposição Lagrangiana, em conjunto com heurísticas propostas, é utilizada para resolver de forma exata o Problema de Rotulação Cartográfica de Pontos (PRCP). Nesse caso, é considerada uma abordagem baseada no Problema do Máximo Conjunto Independente de Vértices (PMCIIV), cujo campo de aplicação é extenso e a formulação linear inteira conhecida, e outra baseada na idéia de rotular o maior número de pontos sem conflitos, para a qual é proposta uma formulação linear inteira, desconhecida até então. As abordagens apresentadas são capazes de provar a otimalidade, algumas até então desconhecidas, para praticamente todas as instâncias utilizadas, e os resultados obtidos superam os apresentados em trabalhos recentes referentes ao PRCP.¹

¹A pesquisa apresentada nesta tese contou com o apoio financeiro parcial da FAPESP (processo 04/11053-9).

NEW APPROACHES FOR MODELING AND FINDING BOUNDS AND SOLUTIONS TO SOME COMBINATORIAL OPTIMIZATION PROBLEMS

ABSTRACT

This thesis presents new strategies for modeling and finding solutions to four different Combinatorial Optimization (CO) problems. In addition, some alternatives to verify the quality of existents solutions are also presented. Firstly, a new approach for modeling a vehicle routing problem known as Dial-a-Ride (DARP) is proposed and a Simulated Annealing (SA) based method is applied to find solutions for it. The proposed model suggests a flexibility on choosing the mainstream optimized by a multi-objective function which can deal with different features of the problem. The proposed method got excellent results and comparing against others found in the literature presents, in all cases, best solutions under a practical point of view. On sequence, the same approach proposed for the DARP is adapted and applied to the Berth Allocation Problem (BAP) found on port management. The Population Training Algorithm (PTA) and its integration with the Linear Programming (LP) by the Column Generation (PTA/LP) are also applied to BAP. The quality of the found solutions is verified by using a commercial software to improve these solutions as initial ones. In addition, the results are still compared with a recent approach in the literature presenting better solutions. Several approaches aiming at bounds for the Unconstrained Binary Quadratic Programming Problem (QP) are also presented. New alternatives of Lagrangian Relaxation with Clusters and Lagrangian Decomposition are applied to QP. The Dantzig-Wolfe Decomposition is also applied with these methods for finding solutions and bounds. The results show that the dual bounds are better than those submitted by traditional lagrangian relaxations and the solutions are better than those presented in some previous work found in the literature. Finally, the Lagrangian Decomposition with some proposed heuristics is used to solve on exact the Point-Feature Cartographic Label Placement Problem (PFCLP). In this case, an approach based on the Maximum Vertex Independent Set Problem (MVISP) is considered. This problem has an extensive application area and an integer linear model are known. Another approach for the PFCLP is based on labelling the largest number of conflicts free points wich an integer linear model is unknown. Suggested approaches are able to prove the optimality (some hitherto unknown) for practically all the used instances and the results improve the ones presented in recent studies concerning the PFCLP. ²

²The research presented in this thesis had a partial financial support of FAPESP (process 04/11053-9).

SUMÁRIO

Pág.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

LISTA DE SÍMBOLOS

1 INTRODUÇÃO	29
1.1 Organização do restante da tese	31
2 FUNDAMENTOS TEÓRICOS	33
2.1 Otimização multi-objetivo	33
2.2 Programação Linear	34
2.3 Decomposição Dantzig-Wolfe	35
2.4 Geração de Colunas	36
2.5 <i>Simulated Annealing</i>	39
2.6 Algoritmos Genéticos	40
2.6.1 Algoritmo Genético Construtivo	41
2.6.2 Algoritmo de Treinamento Populacional	41
2.7 ATP/PL	43
2.8 Relaxações	46
2.8.1 Relaxação Lagrangiana	47
2.8.2 Relaxação Lagrangiana com <i>Clusters</i>	48
2.8.3 Decomposição Lagrangiana	51
2.9 Algoritmo de Subgradientes	52
3 PROBLEMA <i>DIAL-A-RIDE</i>	55
3.1 Revisão da literatura	57
3.1.1 Abordagens clássicas	57
3.1.2 Abordagens recentes	59
3.2 Modelos existentes	61
3.2.1 Modelo proposto por Cordeau (2006)	63

3.2.2	Modelo proposto por Bergvinsdottir (2004)	65
3.3	Abordagem voltada para a qualidade do serviço	68
3.4	<i>Simulated Annealing</i> aplicado ao DARP	72
3.4.1	Geração da solução inicial	72
3.4.2	Estrutura de vizinhança	74
3.5	Experimentos computacionais	77
3.5.1	CPLEX	78
3.5.2	<i>Simulated Annealing</i>	78
3.6	Considerações finais	82
4	PROBLEMA DE ALOCAÇÃO DE BERÇOS	85
4.1	Revisão da literatura	86
4.2	Formulação matemática	88
4.3	Modelo proposto	91
4.4	<i>Simulated Annealing</i> aplicado ao PAB	92
4.4.1	Geração da solução inicial	92
4.4.2	Estrutura de vizinhança	93
4.5	ATP aplicado ao PAB	96
4.6	ATP/PL aplicado ao PAB	98
4.7	Experimentos computacionais	102
4.7.1	Soluções iniciais para o CPLEX	108
4.8	Considerações finais	109
5	PROBLEMA DE PROGRAMAÇÃO QUADRÁTICA BINÁRIA IRRESTRITA	111
5.1	Revisão da literatura	111
5.2	Formulação matemática	114
5.3	Relaxações Lagrangianas tradicionais	115
5.4	Outras relaxações	116
5.5	Relaxações Lagrangianas com <i>Clusters</i>	117
5.5.1	1ª abordagem	118
5.5.2	2ª abordagem	119
5.5.3	Análise das abordagens propostas	120
5.6	Decomposições Lagrangianas	120
5.6.1	1ª abordagem	121
5.6.2	2ª abordagem	123

5.6.3	3ª abordagem	124
5.6.4	Análise das abordagens propostas	127
5.7	Geração de Colunas	127
5.7.1	GC aplicada à LagClus	127
5.7.2	GC aplicada à DecLag	131
5.8	Utilização de uma posiforme	133
5.8.1	LagClus aplicada ao PPQL	135
5.8.2	DecLag aplicada ao PPQL	136
5.9	Experimentos computacionais	137
5.10	Considerações finais	155
6	PROBLEMA DE ROTULAÇÃO CARTOGRÁFICA DE PONTOS	159
6.1	Revisão da literatura	162
6.2	Decomposição do PRCP em <i>clusters</i>	164
6.3	DecLag para o PRCP como um PMCIV	166
6.4	DecLag para o PRCP como um PMNRSC	169
6.5	Experimentos computacionais	174
6.6	Considerações finais	181
7	CONCLUSÕES E TRABALHOS FUTUROS	183
7.1	Trabalhos futuros	185
	REFERÊNCIAS BIBLIOGRÁFICAS	187
A	APÊNDICE - EXEMPLO DE APLICAÇÃO DAS ABORDA-	
	GENS PROPOSTAS PARA O PQ	201
A.1	Limitantes para o PQ	204
A.2	Relaxação Lagrangiana com <i>Clusters</i>	207
A.2.1	1ª abordagem	208
A.2.2	2ª abordagem	209
A.3	Decomposição Lagrangiana	210
A.3.1	1ª abordagem	210
A.3.2	2ª abordagem	213
A.3.3	3ª abordagem	216
A.4	Geração de Colunas	221
A.4.1	GC aplicada à LagClus	221
A.4.2	GC aplicada à DecLag	226

A.5	Formulação posiforme	233
A.5.1	LagClus aplicada ao PPQL	236
A.5.2	DecLag aplicada ao PPQL	238

LISTA DE FIGURAS

	<u>Pág.</u>
2.1 Algoritmo <i>Simulated Annealing</i>	39
2.2 Algoritmo de Treinamento Populacional.	42
2.3 Algoritmo ATP/PL.	45
2.4 Particionamento de um grafo na LagClus.	49
2.5 Decomposição do problema \bar{P}	51
2.6 Algoritmo de Subgradientes.	53
3.1 Casos distintos do DARP.	56
3.2 Heurística de distribuição para o DARP.	72
3.3 Heurística de programação para o DARP.	73
3.4 Movimento re-ordenar rota.	75
3.5 Movimento re-alocar pontos.	75
3.6 Movimento trocar pontos.	76
3.7 Algoritmo <i>Simulated Annealing</i> implementado para o DARP.	77
4.1 Cenário típico para o PAB.	85
4.2 Variáveis referentes ao tempo.	89
4.3 Heurística de distribuição para o PAB.	93
4.4 Heurística de programação para o PAB.	93
4.5 Movimento re-ordenar navios.	94
4.6 Movimento re-alocar navios.	95
4.7 Movimento trocar navios.	95
4.8 Representação de um indivíduo no ATP.	96
4.9 Cruzamento utilizado no ATP.	97
4.10 Representação de um indivíduo no ATP/PL.	99
4.11 Heurística de treinamento utilizada no ATP/PL.	101
4.12 Mutação utilizada no ATP/PL.	101
4.13 Operador de cruzamento utilizado no ATP/PL.	102
5.1 Vértices e arestas necessárias para formação da restrição de corte.	116
5.2 Formação do PMR inicial na Geração de Colunas com a LagClus.	130
5.3 Passos para execução da Geração de Colunas com a LagClus.	130
5.4 Grafo gerado pelo modelo PPQ.	134
6.1 Exemplo de um mapa com rótulos sobrepostos.	159

6.2	Padronização cartográfica proposta por Christensen et al. (1995).	160
6.3	Grafo de conflitos para o PRCP.	161
6.4	Decomposição do PRCP em <i>clusters</i> .	164
6.5	Heurística lagrangiana para o PMCIV.	169
6.6	Exemplo de grafo e restrições para o PMNRSC.	170
6.7	Heurística lagrangiana para o PMNRSC.	173
A.1	Grafo para a formulação do PQ.	201
A.2	Formulação do PQ.	201
A.3	Grafo para a formulação do PQ com uma matriz simétrica.	202
A.4	Formulação do PQ com uma matriz simétrica.	202
A.5	Formulação linear do PQ (PQL).	203
A.6	Relaxação linear do PQL (<i>roof dual</i>).	204
A.7	Formulação do PQLC.	205
A.8	Modelo da Relaxação Lagrangiana da restrição 5.4 do PQL.	206
A.9	Modelo da Relaxação Lagrangiana da restrição 5.6 do PQL.	206
A.10	Particionamento do grafo nas abordagens da LagClus.	207
A.11	Modelo PQL^2 utilizado nas abordagens da LagClus.	207
A.12	Modelo $LC_{\alpha\lambda}PQL^2$ gerado na 1ª abordagem da LagClus.	208
A.13	Subproblemas gerados na 1ª abordagem da LagClus.	208
A.14	Modelo $LC_{\beta\lambda}PQL^2$ gerado na 2ª abordagem da LagClus.	209
A.15	Subproblemas gerados na 2ª abordagem da LagClus.	209
A.16	Particionamento do grafo na 1ª abordagem da DecLag.	210
A.17	Modelo PQL^2 utilizado na 1ª abordagem da DecLag.	211
A.18	Modelo $DC_{\alpha\beta}PQL^2$ gerado na 1ª abordagem da DecLag.	212
A.19	Subproblemas gerados na 1ª abordagem da DecLag.	212
A.20	Particionamento do grafo na 2ª abordagem da DecLag.	213
A.21	Modelo PQL^2 utilizado na 2ª abordagem da DecLag.	214
A.22	Modelo $DC_{\alpha}PQL^2$ gerado na 2ª abordagem da DecLag.	215
A.23	Subproblemas gerados na 2ª abordagem da DecLag.	215
A.24	Formulação linear do PQ considerando toda a matriz \mathbf{Q} (PQLT).	216
A.25	Particionamento do grafo na 3ª abordagem da DecLag.	217
A.26	Modelo $PQLT^2$ utilizado na 3ª abordagem da DecLag.	218
A.27	Modelo $DC_{\alpha\beta}PQLT^2$ gerado na 3ª abordagem da DecLag.	219
A.28	Subproblemas gerados na 3ª abordagem da DecLag.	220
A.29	Vetores utilizados na GC com a LagClus.	221
A.30	Matriz referente às restrições relaxadas na GC com a LagClus.	221
A.31	Matriz referente às restrições <i>intra-cluster</i> na GC com a LagClus.	221

A.32	Representação matricial de PQL^2 na GC com a LagClus.	222
A.33	Subproblema Z_1^{LC} na GC com a LagClus.	222
A.34	Subproblema Z_2^{LC} na GC com a LagClus.	223
A.35	Exemplo de PMR inicial para a GC com a LagClus.	225
A.36	Vetores utilizados na GC com a DecLag.	226
A.37	Matrizes referentes às restrições relaxadas na GC com a DecLag.	226
A.38	Matrizes referentes às restrições <i>intra-cluster</i> na GC com a DecLag.	227
A.39	Representação matricial de $PQLT^2$ na GC com a DecLag.	228
A.40	Subproblema Z_1^{DC} na GC com a DecLag.	229
A.41	Subproblema Z_2^{DC} na GC com a DecLag.	230
A.42	Exemplo de PMR inicial para a GC com a DecLag.	232
A.43	Formulação baseada na posiforme (PPQ).	233
A.44	Grafo baseado nos termos da função objetivo do PPQ.	233
A.45	Grafo para o PPQ baseado nas variáveis y_i	234
A.46	Formulação linear do PPQ (PPQL).	234
A.47	Formulação do PPQL com a eliminação de vértices desconectados.	235
A.48	Grafo para a formulação do PPQL sem os vértices desconectados.	235
A.49	Particionamento do grafo baseado na posiforme com a LagClus.	236
A.50	Modelo $PPQL^2$ utilizado na LagClus.	236
A.51	Modelo $LC_\alpha PPQL^2$ gerado na LagClus.	237
A.52	Subproblemas gerados para a posiforme na LagClus.	237
A.53	Particionamento do grafo baseado na posiforme com a DecLag.	238
A.54	Modelo $PPQL^2$ utilizado na DecLag.	238
A.55	Modelo $DC_\alpha PPQL^2$ gerado na DecLag.	239
A.56	Subproblemas gerados para a posiforme na DecLag.	239

LISTA DE TABELAS

	<u>Pág.</u>	
3.1	Resumo dos experimentos realizados para o DARP.	79
3.2	Melhores soluções obtidas pelo <i>Simulated Annealing</i> para o DARP. . . .	80
3.3	Melhores soluções obtidas pela Busca Tabu para o DARP.	81
3.4	Melhores soluções obtidas pelo Algoritmo Genético para o DARP.	81
4.1	Resultados obtidos pelo SA para o PAB.	103
4.2	Resultados obtidos pelo ATP para o PAB.	104
4.3	Resultados obtidos pelo ATP/PL para o PAB.	105
4.4	Comparação entre os métodos propostos para o PAB.	106
4.5	Comparação com a literatura e com o CPLEX.	107
4.6	Melhores soluções utilizadas como entrada para o CPLEX.	108
5.1	Resultados obtidos pelas relaxações lineares do PQL.	139
5.2	<i>Gaps</i> (%) obtidos pelas relaxações lagrangianas tradicionais.	141
5.3	Resultados obtidos pelas abordagens baseadas na LagClus.	142
5.4	Resultados obtidos pelas abordagens baseadas na DecLag.	144
5.5	<i>Gaps</i> (%) obtidos pelas abordagens baseadas na GC.	146
5.6	<i>Gaps</i> (%) obtidos pelas abordagens baseadas na formulação posiforme. .	147
5.7	Comparação entre os modelos PQL e PPQL.	149
5.8	Comparação entre os <i>gaps</i> (%) para as instâncias das classe A, B e C. . .	150
5.9	Comparação entre os <i>gaps</i> (%) para as instâncias das classe D, E e F. . .	151
5.10	Soluções obtidas para as instâncias das classes B e F.	153
5.11	Soluções obtidas para as instâncias da classe D.	154
5.12	Soluções obtidas para as instâncias das classes A, C e E.	154
6.1	Soluções obtidas pelo PMCIV para as instâncias com 13206 pontos. . . .	175
6.2	Soluções obtidas pelo PMNRSC para as instâncias com 500 pontos. . . .	177
6.3	Soluções obtidas pelo PMNRSC para as instâncias com 750 pontos. . . .	179
6.4	Soluções obtidas pelo PMNRSC para as instâncias com 1000 pontos. . .	180
6.5	Comparação dos resultados obtidos para o PMNRSC com a literatura. . .	181

LISTA DE ABREVIATURAS E SIGLAS

AG	–	Algoritmo Genético
AGC	–	Algoritmo Genético Construtivo
ATP	–	Algoritmo de Treinamento Populacional
ATP/PL	–	Algoritmo ATP/PL
DARP	–	Problema <i>Dial-a-Ride</i>
DecLag	–	Decomposição Lagrangiana
GC	–	Geração de Colunas
LagClus	–	Relaxação Lagrangiana com <i>Clusters</i>
OC	–	Otimização Combinatória
PAB	–	Problema de Alocação de Berços
PL	–	Programação Linear
PLI	–	Programação Linear Inteira
PM	–	Problema Mestre
PMCIIV	–	Problema do Máximo Conjunto Independente de Vértices
PMNRSC	–	Problema do Máximo Número de Rótulos Sem Conflitos
PMR	–	Problema Mestre Restrito
PPC	–	Problema de Particionamento de Conjuntos
PPQ	–	Formulação posiforme do PQ
PPQL	–	Formulação posiforme do PQL
PQ	–	Problema de Programação Quadrática Binária Irrestrita
PQL	–	Formulação linearizada do PQ
PQLC	–	Formulação PQL com a restrição de corte
PRCP	–	Problema de Rotulação Cartográfica de Pontos
SA	–	<i>Simulated Annealing</i>

LISTA DE SÍMBOLOS

α, β, λ	–	Multiplicadores lagrangianos (na maioria dos casos)
δ_k	–	<i>Rank</i> do indivíduo k
Φ	–	Tamanho da população inicial
$\lambda_{k,t}$	–	Variável de decisão nos PMRs
π, μ	–	Variáveis duais
θ_k	–	Custo reduzido da coluna k
τ	–	Probabilidade de mutação
ω	–	Coefficientes de penalização
ψ	–	Tamanho da vizinhança
$\Psi(S)$	–	Vizinhança da solução S
$DC_\alpha P$	–	Decomposição Lagrangiana (em α) para o problema P
$DDC_\alpha P$	–	Dual lagrangiano de $DC_\alpha P$
$DLC_\alpha P$	–	Dual lagrangiano de $LC_\alpha P$
$L_\alpha P$	–	Relaxação Lagrangiana (em α) para o problema P
$LC_\alpha P$	–	Relaxação Lagrangiana com <i>Clusters</i> (em α) para o problema P
PMR_{PL}^{DC}	–	Relaxação Linear do Problema Mestre Restrito para a DecLag
PMR_{PLI}^{DC}	–	Formulação Linear Inteira do Problema Mestre Restrito para a DecLag
PMR_{PL}^{LC}	–	Relaxação Linear do Problema Mestre Restrito para a LagClus
PMR_{PLI}^{LC}	–	Formulação Linear Inteira do Problema Mestre Restrito para a LagClus
$v(P)$	–	Valor da solução do problema P

1 INTRODUÇÃO

Problemas de Otimização Combinatória (OC) são comuns em diversas áreas e estão presentes em grande parte das atividades cotidianas, como os serviços de transporte público, a criação de horários escolares, a confecção de mapas cartográficos, o transporte de crianças para escolas, os serviços de entrega em geral (correios, por exemplo), o controle de tráfego aéreo, etc.

Muitos desses problemas podem ser representados como uma minimização ou maximização de uma função matemática cujas variáveis devem obedecer a certas restrições. Encontrar soluções ótimas ou mesmo aproximadas para esses problemas é um desafio nem sempre fácil de ser vencido. Para alguns desses problemas, são conhecidos métodos eficientes de solução. Já para outros, principalmente para problemas com dimensões reais, métodos de enumeração implícita e relaxações são alguns dos aplicados com maior sucesso.

Logo, percebe-se que para boa parte dos problemas de interesse real, com aplicações práticas, a busca por soluções ótimas por meio de processos enumerativos torna-se inviável, principalmente pelo tempo necessário para se testar todas as possíveis soluções. Nesses casos, várias alternativas vêm sendo desenvolvidas e/ou aprimoradas. Dentre essas, destacam-se aquelas baseadas em modelos matemáticos simplificados e métodos exatos de enumeração implícita, heurísticas e metaheurísticas, e relaxações e decomposições do problema original.

A determinação de modelos matemáticos para esses problemas é uma importante estratégia não só para formalizá-los como também para resolvê-los (principalmente por meio de *softwares* comerciais), ou em alguns casos, para perceber que sua resolução é mais complicada do que parece. Além disso, em geral esses modelos oferecem os subsídios necessários para se propor métodos que explorem o espaço de possíveis soluções de forma mais eficiente, evitando assim a busca exaustiva por boas soluções.

Já as heurísticas e metaheurísticas frequentemente são exploradas quando não se conhece um modelo matemático para um problema, ou quando já foi percebido que “a solução é realmente mais complicada do que parece”. Apesar de não garantir a obtenção das soluções ótimas para os problemas, essas técnicas geralmente apresentam boas soluções em pequenos intervalos de tempo computacional.

Por fim, as relaxações e decomposições apresentam a vantagem de definir valores

limitantes para a possível solução ótima de um problema, e ainda pode apresentar uma informação dual de boa qualidade, o que permite avaliar a proximidade da melhor solução encontrada em relação à solução ótima do problema. Algumas dessas abordagens podem utilizar também a idéia de “dividir para conquistar”, ou seja, decompor um problema muito grande em vários problemas pequenos.

A diversidade de possibilidades para o tratamento de problemas de OC é evidente, assim como a busca por novas alternativas cada vez mais eficientes para resolvê-los. Com esse intuito, busca-se nesta tese explorar um pouco de cada uma dessas principais alternativas. Para isso, são aplicadas e propostas diversas abordagens para o tratamento direto de quatro problemas com grande relevância teórica e prática. As contribuições desta tese são relacionadas a seguir de acordo com os problemas abordados.

- Problema *Dial-a-Ride* (DARP): É proposto um modelo multi-objetivo que flexibiliza o tratamento de casos distintos do problema, principalmente permitindo a busca por soluções que priorizem a qualidade do serviço prestado, e não somente os custos operacionais. O modelo ainda generaliza outros já existentes. É proposta também uma aplicação do *Simulated Annealing* juntamente com outras heurísticas específicas ao DARP para resolver esse modelo.
- Problema de Alocação de Berços (PAB): É proposta uma relaxação de um modelo já existente para representar o PAB. São propostas também aplicações do *Simulated Annealing*, do Algoritmo de Treinamento Populacional e do ATP/PL para resolver esse modelo. É apresentada também uma alternativa, ainda não mencionada na literatura, para verificar a qualidade das soluções obtidas.
- Problema de Programação Quadrática Binária Irrestrita (PQ): É proposta a utilização de uma restrição de corte para fortalecer uma relaxação linear na busca por melhores limitantes para o PQ. São apresentadas duas abordagens distintas de aplicação da Relaxação Lagrangiana com *Clusters* (LagClus) e três baseadas na Decomposição Lagrangiana (DecLag) para obtenção de limitantes para um modelo linear do PQ. A melhor abordagem para cada um desses casos ainda é aplicada em conjunto com a técnica de Geração de Colunas (GC), agora com intuito de também encontrar so-

luções viáveis para o PQ. Por fim, a LagClus e a DecLag são aplicadas a um modelo baseado no PMCIV, resultante de uma transformação baseada em uma posiforme do modelo PQ.

- Problema de Rotulação Cartográfica de Pontos (PRCP): É proposta a aplicação da DecLag, juntamente com heurísticas lagrangianas, para resolver de forma exata duas abordagens distintas para o PRCP. A primeira abordagem considera o PRCP como um Problema do Máximo Conjunto Independente de Vértices (PMCIIV), e a segunda como um Problema do Máximo Número de Rótulos Sem Conflitos (PMNRSC). Nesse último caso, é proposta também uma formulação linear inteira, inexistente até então.

1.1 Organização do restante da tese

O Capítulo 2 apresenta brevemente os principais fundamentos necessários à compreensão das abordagens apresentadas nos capítulos seguintes. Nesse capítulo, é exposta uma idéia geral sobre otimização multi-objetivo e programação linear. Além disso, são apresentadas também as idéias fundamentais referentes às metaheurísticas *Simulated Annealing*, Algoritmos Genéticos, Algoritmo Genético Construtivo e Algoritmo de Treinamento Populacional. São expostas também as idéias da Decomposição Dantzig-Wolfe, da técnica de Geração de Colunas, do método ATP/PL e do Algoritmo de Subgradientes. Por fim, são apresentados alguns conceitos básicos sobre relaxações e suas formas de aplicação.

No Capítulo 3 é abordado o problema *Dial-a-Ride* (DARP). Uma descrição do problema e uma breve revisão bibliográfica referente ao DARP são apresentadas. Além disso, são descritos dois modelos existentes para o DARP e o modelo proposto. Em seguida, é apresentado o método proposto para resolução do problema, os resultados computacionais obtidos e algumas considerações.

O Capítulo 4 apresenta o Problema de Alocação de Berços (PAB). É apresentada uma breve revisão da literatura, e na sequência são apresentados os métodos utilizados para resolver o PAB. Por fim, são apresentados os experimentos computacionais realizados e algumas considerações.

No Capítulo 5 é apresentado o Problema de Programação Quadrática Binária Irrestrita (PQ). São mencionados alguns dos principais trabalhos referentes ao PQ, e são apresentadas as abordagens propostas, baseadas na Relaxação Lagrangiana com

Clusters, na Decomposição Lagrangiana, e na Geração de Colunas. Além disso, são apresentadas algumas relaxações lagrangianas tradicionais para o PQ, uma relaxação linear, e uma relaxação linear com a utilização de uma restrição de corte. Ainda é apresentada uma forma alternativa para representação do problema que utiliza a idéia de posiforme. Finalmente, são apresentados alguns resultados computacionais e algumas considerações.

O Capítulo 6 aborda o Problema de Rotulação Cartográfica de Pontos (PRCP). O PRCP e os trabalhos relacionados mais recentes são descritos brevemente. Na sequência, são tratadas duas abordagens para o problema, uma mais geral, baseada no PMCIV, e outra mais específica, baseada na idéia do PMNRSC. Para o primeiro caso, uma alternativa de Decomposição Lagrangiana é aplicada a um modelo já conhecido para o PMCIV. Já para o segundo caso (PMNRSC), não se conhece um modelo baseado em Programação Linear Inteira (PLI) para representar o PRCP. Logo, é proposto um modelo de PLI, e a Decomposição Lagrangiana é aplicada para resolvê-lo. São apresentados resultados computacionais para as duas abordagens tratadas, e algumas considerações encerram o capítulo.

Finalmente, o Capítulo 7 sumariza as conclusões e contribuições da pesquisa apresentada nesta tese, seguidas por algumas sugestões para sua continuação. O Apêndice A apresenta um exemplo de aplicação das técnicas propostas para o PQ, descritas no Capítulo 5.

2 FUNDAMENTOS TEÓRICOS

Com o intuito de fornecer um embasamento teórico a respeito das metodologias utilizadas ao longo deste trabalho, este capítulo apresenta brevemente os principais fundamentos necessários à compreensão das abordagens apresentadas nos capítulos seguintes.

2.1 Otimização multi-objetivo

Em grande parte dos “problemas de otimização”, o objetivo normalmente aborda a minimização ou maximização de um custo bem específico (geralmente referente a custos operacionais). Entretanto, existe um número expressivo de problemas que apresentam mais de um objetivo. Esse tipo de problema pode ser classificado como *problema de otimização multi-objetivo*, cujo objetivo final é formado pela otimização de mais de uma função objetivo.

Como definido em Cohon (1978) e Steuer (1986), um problema multi-objetivo consiste em determinar um vetor de variáveis de decisão que otimize um vetor de funções objetivo e satisfaça o conjunto de restrições envolvidas. Uma função multi-objetivo pode ser representada matematicamente da seguinte maneira:

$$\text{Minimizar } f(S) = \begin{bmatrix} f_1(S) \\ f_2(S) \\ \vdots \\ f_\sigma(S) \end{bmatrix} \quad (2.1)$$

$f_i(S)$, $i = 1, \dots, \sigma$ são as σ funções objetivo para esse problema. Não é comum o fato de que os diferentes objetivos possam ser otimizados simultaneamente com os mesmos valores de parâmetros. Então, algum tipo de “troca” entre os critérios das funções objetivos é necessário para assegurar a obtenção de resultados satisfatórios.

O conceito de “otimalidade” não se aplica diretamente a problemas multi-objetivo. Nesse caso, tal conceito é substituído pela noção de “Pareto Ótimo”. Basicamente, um vetor S^* é denominado *Pareto Ótimo* para a função multi-objetivo 2.1 se todos os outros vetores S' possuírem valores “mais altos” (em problemas de minimização) para pelo menos uma das funções objetivos $f_i(S')$. Ou seja, uma solução S é chamada *Pareto Ótima* (ou eficiente ou não-dominada) se o valor de alguma função objetivo $f_i(S')$ não puder ser melhorado sem piorar ao menos uma das outras.

Os problemas multi-objetivo são geralmente resolvidos por meio da combinação dos múltiplos objetivos em um objetivo único. A solução desse objetivo único é um ponto no pareto ótimo para a função multi-objetivo original.

Uma técnica usada frequentemente (JORGENSEN et al., 2007; MAURI, 2003; MAURI, 2005; MAURI; LORENA, 2006a; MAURI; LORENA, 2007) para combinar os múltiplos objetivos de um problema multi-objetivo é a minimização (ou maximização) do somatório dos vários objetivos, sendo cada um desses penalizado com um valor positivo (ou negativo, no caso de um problema de maximização). A função a seguir apresenta uma combinação para o problema descrito na função 2.1. A escolha dos valores dos pesos ω_i é baseada na “importância” de cada uma das diferentes funções objetivo, de acordo com a avaliação do usuário.

$$\text{Minimizar } \sum_{i=1}^{\sigma} \omega_i f_i(S) \quad \omega_i > 0, \quad i = 1, 2, \dots, \sigma \quad (2.2)$$

É possível tratar uma função multi-objetivo de outras maneiras, como por exemplo, pela programação *multi-nível*. Na programação *multi-nível* os objetivos são ordenados pela importância, e a partir de então, um conjunto com os pontos que otimizam o objetivo mais importante é encontrado. Feito isso, os pontos que pertencem a esse conjunto são otimizados, e o segundo objetivo mais importante é encontrado, e assim por diante, até todos os objetivos serem otimizados, resultando em conjuntos cada vez menores. Mais detalhes dessa técnica são apresentados em Migdalas et al. (1997).

2.2 Programação Linear

Um problema de Programação Linear (PL) consiste em minimizar ou maximizar uma função linear considerando um conjunto de restrições formado por equações e/ou inequações lineares (BAZARAA et al., 1990). Caso as variáveis de decisão desse problema sejam estritamente inteiras, a Programação Linear é chamada de Programação Linear Inteira (PLI), ou simplesmente Programação Inteira (WOLSEY, 1998).

O método Simplex, que é uma das principais formas de se resolver um problema de PL, foi desenvolvido em 1947 por G. B. Dantzig, e apresentado em 1963 em um de seus mais famosos livros, *Linear Programming and Extensions* (DANTZIG, 1963).

A PL está entre as técnicas de maior sucesso e mais amplamente aplicadas em problemas de otimização, e é utilizada, com sucesso, em praticamente todos os tipos de problemas combinatórios, como por exemplo, problemas de geração de escalas, geração de quadros de horários, roteamento de veículos, etc. Mais detalhes dessa técnica são apresentados em Bazaraa et al. (1990), Dantzig (1963) e Wolsey (1998).

2.3 Decomposição Dantzig-Wolfe

A Decomposição Dantzig-Wolfe é uma técnica que decompõe um problema de PL permitindo que sua resolução seja dada por meio de soluções alternadas para subproblemas lineares (que representam partes do problema original) e de um problema coordenador obtido a partir de transformações lineares desses subproblemas (DANTZIG; WOLFE, 1960). Uma descrição formal dessa técnica, baseada nas descrições apresentadas em Ribeiro (2007) e Wolsey (1998), é apresentada a seguir.

Considerando um problema $P: v(P) = \mathbf{Max}\{cx : x \in X\}$ com uma região factível X que pode ser descrita como a interseção de dois ou mais conjuntos com a estrutura $X = \bigcap_{k=1}^K X^k$, para algum $K > 1$, esse problema P pode ser reescrito da seguinte forma:

$$P : \quad v(P) = \mathbf{Maximizar:}$$

$$\sum_{k=1}^K c^k x^k \tag{2.3}$$

Sujeito a:

$$\mathbf{A}^1 x^1 + \mathbf{A}^2 x^2 + \dots + \mathbf{A}^K x^K = b \tag{2.4}$$

$$\begin{aligned} \mathbf{D}^1 x^1 &\leq d_1 \\ \dots &\leq d_2 \\ \dots &\leq \vdots \\ \mathbf{D}^K x^K &\leq d_K \end{aligned} \tag{2.5}$$

$$x^1 \in Z_+^{n_1}, x^2 \in Z_+^{n_2}, \dots, x^k \in Z_+^{n_K} \tag{2.6}$$

Nesse modelo, x^k ($k = 1, \dots, K$) são variáveis de decisão e c^k ($k = 1, \dots, K$) seus respectivos coeficientes (custos); b, d_1, d_2, \dots, d_K indicam a disponibilidades de recursos; e \mathbf{A}^k e \mathbf{D}^k ($k = 1, \dots, K$) são matrizes de coeficientes. Os conjuntos

$X^k = \{x^k \in Z_+^{n_k} : \mathbf{D}^k x^k \leq d_k\}$ devem ser independentes para $k = 1, \dots, K$, e somente a restrição 2.4 deverá acoplar os diferentes conjuntos de variáveis.

Considerando que cada conjunto X^k possua um conjunto finito T_k de soluções (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:

$$\left\{ x^k \in \mathbb{R}^{n_k} : x^k = \sum_{t=1}^{T_k} \lambda_{k,t} x^{k,t}, \sum_{t=1}^{T_k} \lambda_{k,t} = 1, \lambda_{k,t} \in [0, 1], \forall t = 1, \dots, T_k \right\} \quad (2.7)$$

Logo, usando essa definição de X^k , o problema P pode ser assim definido:

$P : \quad v(P) = \text{Maximizar:}$

$$\sum_{k=1}^K \sum_{t=1}^{T_k} (c^k x^{k,t}) \lambda_{k,t} \quad (2.8)$$

Sujeito a:

$$\sum_{k=1}^K \sum_{t=1}^{T_k} (A^k x^{k,t}) \lambda_{k,t} \leq b \quad (2.9)$$

$$\sum_{t=1}^{T_k} \lambda_{k,t} = 1 \quad \forall k = 1, \dots, K \quad (2.10)$$

$$\lambda_{k,t} \in [0, 1] \quad \forall t = 1, \dots, T_k, k = 1, \dots, K \quad (2.11)$$

O problema de encontrar soluções viáveis (pontos extremos) para X^k é chamado de subproblema, e o problema descrito pelas expressões 2.8 a 2.11 é denominado Problema Mestre (PM). Esses dois problemas definem a Decomposição Dantzig-Wolfe. Mais detalhes dessa técnica são apresentadas em Bazaraa et al. (1990) e Wolsey (1998).

2.4 Geração de Colunas

A Geração de Colunas é uma técnica baseada na idéia de resolver um problema de PL através da adição de variáveis (colunas) durante a fase de *pricing* do método Simplex. Nessa técnica, um Problema Mestre (PM) permite a seleção de um melhor subconjunto de colunas (i.e. variáveis de decisão) e é resolvido por uma formulação de PL do problema. Esse problema, por sua vez, tem seu próprio modelo,

baseado frequentemente em um modelo de cobertura ou particionamento de conjuntos com restrições adicionais, como por exemplo, restrições básicas de cobertura e restrições globais, como o número de recursos disponíveis (veículos, tripulações, caminhões, etc.) (LAGRÈZE; LEBBAR, 2000).

Conhecida há várias décadas pela sua capacidade de resolver problemas grandes de PL, a idéia principal dessa técnica consiste em: considerar um pequeno número de variáveis (colunas) de cada vez; resolver o problema de PL (problema mestre formado por essas colunas) e obter as soluções primal e dual; gerar novas colunas interessantes, com custo reduzido positivo (para problemas de maximização), com o uso da solução dual do problema principal para melhorar a solução da PL anterior; e repetir esse processo até que nenhuma melhora possa ser obtida na solução da PL (LAGRÈZE; LEBBAR, 2000).

Considerando a relaxação linear do problema mestre (PM_{PL}) descrito pelas expressões 2.8 a 2.11, e $\begin{pmatrix} c^k x \\ \mathbf{A}^k x \end{pmatrix}$ uma coluna para cada $x \in X^k$, tem-se as variáveis duais $\{\pi_i\}_{i=1}^m$ associadas as m restrições de acoplamento 2.9, e $\{\mu_i\}_{i=1}^K$ as variáveis duais associadas às restrições de convexidade 2.10.

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 sua enumeração explícita inviabiliza o uso direto do PM. Sendo assim, opta-se frequentemente por trabalhar com um subconjunto de pontos extremos, e em seguida, geram-se novos pontos extremos de maneira sistemática.

Logo, considerando $X^{k,l} \subseteq X^k$ como um subconjunto que representa os l pontos extremos de X^k , o problema mestre (2.8 a 2.11) com apenas um subconjunto de pontos extremos $X^{k,l}$ para cada k , passa a ser denominado Problema Mestre Restrito (PMR). Sendo assim, pode-se generalizar a relaxação linear do PMR da seguinte forma:

$$PMR_{PL} : \quad \tilde{v}(PMR_{PL}) = \text{Maximizar:} \quad \tilde{c}\tilde{\lambda} \quad (2.12)$$

$$\text{Sujeito a:} \quad \tilde{\mathbf{A}}\tilde{\lambda} \leq \tilde{\mathbf{b}} \quad (2.13)$$

$$\tilde{\lambda} \geq 0 \quad (2.14)$$

$\tilde{b} = \begin{pmatrix} b \\ 1 \end{pmatrix}$, $\tilde{\mathbf{A}}$ é uma matriz gerada conforme o número de colunas disponíveis, e \tilde{c} corresponde aos custos referentes a essas colunas.

Ao resolver o PMR_{PL} , obtém-se uma solução ótima $\tilde{\lambda}^*$ do problema, e uma solução ótima do dual $(\pi, \mu) \in \mathbb{R}^m \times \mathbb{R}^K$. Qualquer solução viável do PMR_{PL} é viável para PM_{PL} . Em particular, se $\tilde{\lambda}^*$ é uma solução viável de PM_{PL} , então $\tilde{v}(PMR_{PL}) = \tilde{c}\tilde{\lambda}^* = \sum_{i=1}^m \pi_i b_i + \sum_{k=1}^K \mu_k \leq v(PM_{PL})$ (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. Logo, esse subproblema pode ser definido como:

$$Z_k = \text{Max} \{ (c^k - \pi \mathbf{A}^k)x : x \in X^k \} \quad \forall k = 1, \dots, K \quad (2.15)$$

Desta forma, o subproblema torna-se um problema de custo reduzido θ ($\theta_k = Z_k - \mu_k$) ou problema *pricing* (WOLSEY, 1998). Caso $\theta_k > 0$ para algum k , a solução correspondente à solução ótima \tilde{x}^k do subproblema tem um custo reduzido positivo, a coluna $\begin{pmatrix} c^k \tilde{x}^k \\ \mathbf{A}^k \tilde{x}^k \end{pmatrix}$ deve ser inserida no PMR_{PL} , e o novo PMR deve ser novamente resolvido. Logo, $\theta_k \geq (c^k - \pi \mathbf{A}^k)x - \mu_k$ para todo $x \in X^k$, e conseqüentemente $(c^k - \pi \mathbf{A}^k)x - \mu_k - \theta_k \leq 0$ para todo $x \in X^k$. Assim, sendo $\theta = (\theta_1, \dots, \theta_K)$, tem-se que $(\pi, \mu + \theta)$ é um dual viável para o PM_{PL} . Então, pode-se afirmar que:

$$v(PM_{PL}) \leq \pi b + \sum_{k=1}^K (\mu_k + \theta_k) \quad (2.16)$$

Por fim, se $\theta_j > 0$ (para problemas de maximização) é interessante que a coluna seja introduzida no PMR para tentar melhorar a sua solução (este é o primeiro critério de Dantzig do algoritmo Simplex), e caso $\theta_k \leq 0$ para todo $k = 1, \dots, K$, a geração de colunas é encerrada, pois uma solução ótima foi obtida. Assim, um método eficiente para gerar novas colunas com custo reduzido positivo é gerar essas colunas dinamicamente, considerando os valores das variáveis duais da solução corrente. Mais detalhes dessa técnica são apresentados em Wilhelm (2001) e Wolsey (1998).

2.5 *Simulated Annealing*

O *Simulated Annealing* (SA) é um método de busca local que aceita movimentos de piora para escapar de ótimos locais. O SA foi proposto originalmente por Kirkpatrick et al. (1983) e se fundamenta em uma analogia com a termodinâmica ao simular o resfriamento de um conjunto de átomos aquecidos. Essa técnica começa sua busca a partir de uma solução inicial qualquer. O procedimento principal consiste em um *loop* que gera aleatoriamente, em cada iteração, um único vizinho S' da solução corrente S .

A cada geração de um vizinho S' de S ($S' \in \Psi(S)$), é testada a variação do valor da função objetivo (custo), isto é, $\Delta = f(S') - f(S)$. Para um problema de minimização, se $\Delta < 0$ o método aceita a solução e S' passa a ser a nova solução corrente. Caso $\Delta \geq 0$ a solução vizinha candidata também poderá ser aceita, mas neste caso, com uma probabilidade $e^{-\Delta/T}$, onde T é um parâmetro do método chamado de temperatura, que regula a probabilidade de aceitação de soluções de pior custo.

```
1. DADO ( $\alpha$ , SAmáx,  $T_0$ ,  $T_c$  e  $S$ ) FAÇA
2.  $S^* \leftarrow S$ ;           {Melhor solução obtida até então}
3. IterT  $\leftarrow 0$ ;       {Número de iterações na temperatura T}
4.  $T \leftarrow T_0$ ;        {Temperatura corrente}
5. ENQUANTO ( $T > T_c$ ) FAÇA
6.   ENQUANTO (IterT < SAmáx) FAÇA
7.     IterT  $\leftarrow$  IterT + 1;
8.     GERAR (um vizinho qualquer  $S' \in \Psi(S)$ );
9.      $\Delta \leftarrow f(S') - f(S)$ ;
10.    SE ( $\Delta < 0$ )  $S \leftarrow S'$ ;
11.    SE ( $f(S') < f(S^*)$ )  $S^* \leftarrow S'$ ; FIM-SE;
12.    SENÃO
13.      TOMAR ( $x \in [0, 1]$ );
14.      SE ( $x < e^{-\Delta/T}$ )  $S \leftarrow S'$ ; FIM-SE;
15.    FIM-SE;
16.  FIM-ENQUANTO;
17.   $T \leftarrow \alpha * T$ ;  IterT  $\leftarrow 0$ ;
18. FIM-ENQUANTO;
19.  $S \leftarrow S^*$ ;
20. RETORNAR ( $S$ );
```

Figura 2.1 - Algoritmo *Simulated Annealing*.

Fonte: Adaptada de Mauri (2003, p. 17).

A temperatura T assume inicialmente um valor elevado T_0 . Após um número fixo de iterações SAm_{ax} (o qual representa o número de iterações necessárias para o sistema atingir o equilíbrio térmico em uma dada temperatura), a temperatura é gradativamente diminuída por uma razão de resfriamento α , tal que, em um instante t , $T_t \leftarrow \alpha * T_{t-1}$, sendo $0 < \alpha < 1$. Com esse procedimento, no início, dá-se uma chance maior para escapar de mínimos locais e, à medida que T aproxima-se de zero, o algoritmo comporta-se como um método de descida, uma vez que diminui a probabilidade de se aceitar movimentos de piora ($T \rightarrow 0 \Rightarrow e^{-\Delta/T} \rightarrow 0$).

O procedimento pára quando a temperatura chega a um valor próximo de zero (temperatura de congelamento: T_C) e nenhuma solução que piore o valor da melhor solução é mais aceita, isto é, quando o sistema está estável. A solução obtida quando o sistema encontra-se nesta situação evidencia o encontro de um mínimo local, o que em alguns casos também pode representar um mínimo global.

Os parâmetros de controle do procedimento são a razão de resfriamento α , o número de iterações para cada temperatura SAm_{ax} , a temperatura inicial T_0 , e a temperatura de congelamento T_C . A Figura 2.1 apresenta um algoritmo *Simulated Annealing* “padrão”. Algumas aplicações desse método são apresentadas em Mauri (2003), Mauri e Lorena (2006a), Mauri e Lorena (2006b) e Mauri et al. (2008b).

2.6 Algoritmos Genéticos

Desde os anos 80 os Algoritmos Genéticos (AGs) são utilizados para solucionar problemas de otimização combinatória e de outras áreas. Um AG é um procedimento computacional capaz de emular o processo Darwiniano de seleção natural que vem se mostrando ser bastante interessante em aplicações de OC.

Em analogia com o sistema biológico, cada candidato à solução do problema representa um cromossomo ou indivíduo da população, que é o conjunto de todos esses candidatos. Cada indivíduo da população pode ser representado no AG por um conjunto binário (0 e 1), por números inteiros, por números de ponto flutuante, ou por caracteres. Esses indivíduos ainda têm um custo associado, que determina sua habilidade para sobreviver e produzir descendentes no processo de seleção natural. Para melhorar o custo ou a aptidão dos indivíduos das populações sucessivas são necessários os operadores genéticos. Esses operadores permitem selecionar indivíduos mais aptos à sobrevivência para reprodução, além de manter as características de adap-

tação adquiridas pelas gerações passadas, criando indivíduos cada vez mais aptos. Uma descrição detalhada dessa técnica é apresentada em Goldberg (1989).

2.6.1 Algoritmo Genético Construtivo

O Algoritmo Genético Construtivo (AGC) foi primeiramente abordado por Lorena e Lopes (1996). O aspecto construtivo do AGC está relacionado ao fato de que, ao longo do processo seletivo, uma população de soluções incompletas (esquemas) é utilizada como base para a construção de uma população de soluções completas (estruturas). Tanto esquemas quanto estruturas competem entre si durante o processo evolutivo. Estruturas são formadas a partir de recombinações de esquemas, e então tem-se uma melhora tanto das estruturas quanto dos esquemas ao longo das gerações. Outras características importantes e inovadoras do AGC estão na utilização de populações de tamanho variável ao longo das gerações, e a possibilidade de usar heurísticas para definição das funções de avaliação de aptidão dos indivíduos.

2.6.2 Algoritmo de Treinamento Populacional

Trata-se de uma técnica evolutiva empregada pioneiramente por Oliveira (2002), derivada do Algoritmo Genético Construtivo (AGC), e desenvolvida no Instituto Nacional de Pesquisas Espaciais (INPE). Segundo Oliveira (2002), o termo “treinamento” significa fazer com que certo modelo seja adaptado a um ambiente, inicialmente desconhecido, por meio do ajuste de seus parâmetros de acordo com alguma medida de desempenho esperada.

Na proposta de tese de Oliveira (2002) é apresentado o Algoritmo de Treinamento Populacional (ATP), baseando-se nas idéias do AGC e principalmente no uso de heurísticas para definição das aptidões dos indivíduos na população. O ATP pode ser visto como uma nova forma de direcionar a evolução de indivíduos (soluções candidatas), utilizando-se para isso informações sobre o ambiente (problema) a ser assimilado. Tais informações são extraídas do ambiente por heurísticas específicas. Um indivíduo é considerado bem adaptado de acordo com certa heurística se ele não puder evoluir (ser melhorado) com relação à heurística de treinamento empregada. Um indivíduo bem adaptado deve então participar do máximo de gerações possíveis no processo evolutivo.

A adaptação no treinamento populacional é então utilizada como direcionador, ou seja, regiões do espaço de busca (universo de possíveis combinações de soluções)

ocupadas por indivíduos bem adaptados serão mais exploradas que as demais. A Figura 2.2 apresenta o ATP.

```

1. DADO ( $\Phi, \tau, \kappa, ATP_{max}, g_{max}, Y, \beta, Step$ )
2. CRIAR (uma população  $Pop$  com  $\Phi$  indivíduos);
3. ENQUANTO ( $t \leq ATP_{max}$ )
4.    $t \leftarrow t+1$ ;
5.   SELECIONAR (indivíduo base - entre os  $\kappa$  melhores);
6.   SELECIONAR (indivíduo guia - de toda a população);
7.    $S \leftarrow$  RECOMBINAR (base, guia);
8.   AVALIAR ( $S$ );
9.   APLICAR (mutação em  $S$  - com probabilidade  $\tau$ );
10.  AVALIAR (a vizinhança do novo indivíduo -  $\Psi(S)$ );
11.  CALCULAR ( $\delta(S)$ );
12.  SE ( $\delta(S) > \beta$ )
13.    ATUALIZAR ( $Pop$  - inserir  $S$ );
14.  FIM-SE;
15.  ATUALIZAR ( $\beta$ );
16.  ENQUANTO (existir  $\delta(S) \leq \beta$ )
17.    ELIMINAR ( $S$ );
18.  FIM-ENQUANTO;
19. FIM-ENQUANTO;

```

Figura 2.2 - Algoritmo de Treinamento Populacional.

Fonte: Adaptada de Oliveira e Lorena (2003).

O ATP começa a partir de uma população Pop , com Φ indivíduos, que é gerada aleatoriamente ou por outro método de inicialização. A partir dessa população são criados novos indivíduos a cada geração t , até que um critério de parada seja satisfeito (normalmente um número máximo de gerações). Os novos indivíduos são criados a partir de um cruzamento entre dois indivíduos já existentes na população. Esses dois indivíduos, chamados de “base” e “guia”, são selecionados a partir de uma parte da população (κ melhores) que contém os indivíduos mais adaptados e a partir de toda a população, respectivamente. Os novos indivíduos podem ainda sofrer “mutações” (com probabilidade τ), diversificando assim as características herdadas de seus “ancestrais”.

O novo indivíduo é então avaliado pelas funções g e h , utilizadas para o processo de treinamento evolutivo. A função $g(S)$ mostra o custo que determina a habilidade de sobrevivência do indivíduo S . Para a definição da função $h(S)$, são avaliados vários indivíduos alternativos S' na vizinhança de S ($S' \in \Psi(S)$), tomando-se o melhor (de menor custo - para problemas de minimização), isto é, $h(S) = \text{Melhor } g(S') \mid S' \in \Psi(S)$. O processo evolutivo é desenvolvido privilegiando os indivíduos de menor

diferença $[g(S) - h(S)]$, atribuindo os seguintes valores de *rank* aos indivíduos:

$$\delta(S) = \gamma \times [g_{max} - g(S)] - [g(S) - h(S)] \quad (2.17)$$

g_{max} é um limitante superior de custo alto, e γ um percentual constante de g_{max} , isto é, uma constante de proporcionalidade para o intervalo $[g_{max} - g(S)]$. A população é controlada dinamicamente por um parâmetro de evolução denominado β , que é definido por:

$$\beta_t = \beta_{t-1} + Step \times \Phi_t \times \frac{\delta_{bst} - \delta_{wst}}{RG} \quad (2.18)$$

Step é uma constante que controla a velocidade da evolução, $(\delta_{bst} - \delta_{wst})$ é a variação, no momento, entre os valores dos *ranks* do melhor e do pior indivíduo, respectivamente, e *RG* é o número de gerações que faltam para terminar o processo. Dessa forma, a partir dos valores dos *ranks* (Equação 2.17) é determinada a permanência ou não dos novos indivíduos na população, ou seja, são aceitos apenas indivíduos com $\delta(S) > \beta$, enquanto os demais são descartados. Esse parâmetro também é atualizado a cada geração, eliminando-se no fim de cada uma destas os indivíduos menos adaptados ($\delta(S) \leq \beta$). Os parâmetros de controle do ATP são: o tamanho da população inicial Φ , o percentual para escolha dos indivíduos base κ , a probabilidade de mutação τ , o tamanho da vizinhança ψ , o número máximo de gerações *ATPmax*, a constante *Step*, e o parâmetro de evolução β .

2.7 ATP/PL

O ATP/PL é um método híbrido recente que foi proposto por Mauri (2005) para resolução de um problema conhecido como “Problema de Escalonamento de Tripulações” (MAURI; LORENA, 2004a; MAURI; LORENA, 2004b; MAURI; LORENA, 2007). Esse método é baseado na técnica de Geração de Colunas (ver Seção 2.4), sendo implementada pela interação entre o Algoritmo de Treinamento Populacional (ver Subseção 2.6.2) e a Programação Linear (PL). O ATP e a PL são aplicados de maneira interativa, sendo o ATP, por meio de informações da relaxação de PL, responsável pela geração de boas colunas (baixo custo e boa cobertura das restrições), e a PL pela resolução de um Problema de Particionamento de Conjuntos (PPC) formado por essas colunas. Esse PPC é formulado da seguinte maneira:

Minimizar:

$$\sum_{j=1}^n c_j x_j \quad (2.19)$$

Sujeito a:

$$\sum_{j=1}^n a_{ij} x_j = 1 \quad i = 1, \dots, m \quad (2.20)$$

$$x_j \in \{0, 1\} \quad j = 1, \dots, n \quad (2.21)$$

c_j representa o custo da coluna j ; x_j é igual a 1 se a coluna j pertencer à solução do problema e 0 caso contrário; n é o número de colunas do problema; m é o número de restrições, e a_{ij} é igual a 1 se a restrição i é atendida pela coluna j e 0 caso contrário.

Inicialmente no ATP/PL é criado um conjunto inicial de colunas (aleatoriamente ou por uma heurística qualquer) que deverá formar um PPC semelhante ao descrito anteriormente. A relaxação linear desse PPC é resolvida e os valores obtidos para as variáveis duais são utilizados no ATP para auxiliar na geração de novas colunas. O ATP é implementado de forma análoga à apresentada na Subseção 2.6.2, porém a definição das funções de aptidão dos indivíduos (colunas) deverão guiar o ATP na busca por novas colunas que melhorem a solução do PPC. Assim, a função g (e consequentemente a função h) é definida com o uso de informações das variáveis duais retiradas da resolução da relaxação linear do PPC. Logo, essa troca de informações consolida a interatividade do ATP com a PL, que é a idéia principal dessa técnica. A definição da função g é dada pela seguinte equação:

$$g_j = \begin{cases} \frac{c_j}{\sum_{i=1}^m \pi_i a_{ij}}, \text{ para } \sum_{i=1}^m \pi_i a_{ij} > 0 \\ c_j, \text{ para } \sum_{i=1}^m \pi_i a_{ij} \leq 0 \end{cases} \quad (2.22)$$

c_j é o custo da coluna j e π_i é a variável dual associada à restrição i . Logo, utilizando os conceitos da técnica de GC (ver Seção 2.4), pode-se calcular o custo reduzido da j -ésima coluna, denotado por θ_j , pela equação:

$$\theta_j = c_j - \sum_{i=1}^m \pi_i a_{ij} \quad (2.23)$$

A partir das equações 2.22 e 2.23, pode-se notar que, para custos reduzidos negativos ($\theta_j < 0$), o valor da função g estará dentro do intervalo $[0,1]$, fazendo com que a heurística de treinamento, que define o valor da função h correspondente (melhor g na vizinhança), proporcione uma pequena diferença ($g - h$). Para colunas com custo reduzido positivo ($\theta_j \geq 0$) o valor da função g será igual ao próprio custo da coluna, ou seja, um valor “elevado”. Assim, a população é treinada indiretamente para geração de indivíduos com custo reduzido negativo, o que conseqüentemente evita a geração de um número excessivo de colunas.

Após a execução do ATP, o novo PPC formado pela inserção das novas colunas geradas é resolvido novamente, e o processo é repetido até que um critério de parada seja satisfeito. Finalmente, terminada essa interação, o PPC contendo todas as colunas geradas é resolvido de forma inteira (PLI) pelo *software* CPLEX (ILOG, 2006), por exemplo.

A resolução do PPC de uma forma exata ainda poderá ser árdua, principalmente para problemas de porte mais elevado. Assim, após algum tempo de processamento, um número de iterações, uma quantidade de colunas geradas, ou um *gap* limite (porcentagem da diferença dos valores da melhor PLI e PL), a melhor solução encontrada é tomada como solução para o problema original.

- | |
|--|
| <ol style="list-style-type: none"> 1. <u>CRIAR</u> (conjunto inicial de colunas - formar um PPC); 2. <u>RESOLVER</u> (o PPC através da relaxação de PL); 3. <u>ENQUANTO</u> (o critério de parada não for satisfeito) 4. <u>EXECUTAR</u> (o ATP utilizando as variáveis duais); 5. <u>REMOVER</u> (colunas inválidas); 6. <u>CALCULAR</u> (custo reduzido das novas colunas); 7. <u>ADICIONAR</u> (colunas com custo reduzido negativo); 8. <u>RESOLVER</u> (o novo PPC através da relaxação de PL); 9. <u>FIM-ENQUANTO</u>; 10. <u>RESOLVER</u> (o PPC através da PLI); |
|--|

Figura 2.3 - Algoritmo ATP/PL.

Fonte: Adaptada de Mauri (2005).

Os parâmetros do ATP/PL são os mesmos do ATP, logo, é necessário que se defina apenas o critério de parada e a quantidade de colunas para formar o PPC inicial. A Figura 2.3 apresenta o pseudo-código do ATP/PL. Mais detalhes dessa técnica são apresentados em Mauri (2005), e algumas aplicações são descritas em Mauri e Lorena (2004a), Mauri e Lorena (2004b), Mauri e Lorena (2007) e Mauri et al. (2008a).

2.8 Relaxações

As relaxações de um problema visam tornar a sua resolução menos árdua. Segundo Wolsey (1998), define-se como relaxação de um problema de maximização P , um problema P' tal que cada solução viável de P também é uma solução viável de P' , e o valor da solução do problema P sempre é menor ou igual à solução de P' (em problemas de maximização).

Uma das relaxações mais triviais encontradas na literatura, para problemas inteiros (PLI), é a relaxação linear das variáveis de decisão do modelo. No caso de problemas de maximização, essa relaxação é chamada *roof dual* (ADAMS; DEARING, 1994; BOROS et al., 1990; BOROS et al., 1992).

Outra relaxação também muito utilizada é baseada na extração de um conjunto de restrições “difíceis” da formulação do problema original. Nesses casos, essas restrições geralmente são adicionadas na função objetivo do problema (WOLSEY, 1998).

Dado o seguinte problema P :

$$P: \quad v(P) = \text{Maximizar:} \quad \sum_j c_j x_j \quad (2.24)$$

Sujeito a:

$$Ax \leq b \quad (2.25)$$

$$Dx \leq d \quad (2.26)$$

$$x \in \{0, 1\} \quad (2.27)$$

O *roof dual* pode ser obtido facilmente pela substituição da restrição 2.27 pela restrição $0 \leq x \leq 1$. Considerando agora que esse problema seja de difícil solução devido à restrição 2.26, pode-se relaxar esse modelo por meio da simples extração dessa restrição, e se for o caso, acrescentá-la na função objetivo. Essa é a idéia geral da Relaxação Lagrangiana apresentada a seguir na Subseção 2.8.1.

Deve-se destacar que, ao se resolver qualquer uma dessas relaxações, não se pode garantir que a solução obtida seja uma solução viável (que satisfaça todas as restrições) para o problema original. Nesses casos, a forma utilizada para relaxar o problema irá influenciar diretamente na “qualidade” da solução relaxada obtida. Esse fato ca-

racteriza as relaxações como *fortes* ou *fracas*, sendo a primeira referente a soluções mais próximas à solução do problema original, e a segunda mais distante.

A solução relaxada de um problema representa um valor limite para a solução ótima do problema original, e caso o objetivo deste seja minimizar determinada função, esse valor é chamado de *limitante inferior*, e caso contrário (maximização), *limitante superior*. Essa abordagem possui a vantagem de definir valores limites para a solução ótima de um determinado problema, e pode apresentar uma informação dual de boa qualidade, o que permite avaliar a proximidade da melhor solução (relaxada) encontrada em relação à solução ótima do problema original.

2.8.1 Relaxação Lagrangiana

Uma das formas mais utilizadas de relaxação é baseada na adição das restrições “difíceis” do modelo na função objetivo com a utilização de um vetor de multiplicadores. Essa idéia consiste em tornar problemas difíceis em problemas fáceis, que se tornam complicados apenas por um número relativamente pequeno de restrições.

Os multiplicadores utilizados são chamados de multiplicadores de Lagrange, e conseqüentemente, essa idéia é conhecida como Relaxação Lagrangiana. Esse método foi apresentado no início dos anos 70 em Held e Karp (1970) e Held e Karp (1971). Logo, para o problema P (2.24 a 2.27), pode-se definir sua Relaxação Lagrangiana $L_\alpha P$ como sendo:

$L_\alpha P$: $v(L_\alpha P) = \text{Maximizar:}$

$$\sum_j c_j x_j - \alpha(Dx - d) \quad (2.28)$$

Sujeito a:

$$Ax \leq b \quad (2.29)$$

$$x \in \{0, 1\} \quad (2.30)$$

Nesse modelo, tem-se que, para qualquer $\alpha \geq 0$ o resultado será uma relaxação do problema original P . É interessante destacar que nesse modelo (relaxado) as restrições “difíceis” aparecem na função objetivo como um termo de penalidade baseado nos multiplicadores lagrangianos associados. Esse modelo $L_\alpha P$ é considerado uma Relaxação Lagrangiana do problema original. Como demonstrado por Held e Karp

(1971), pode-se afirmar que, caso as restrições relaxadas sejam do tipo $Dx \geq d$, os multiplicadores lagrangianos devem ser menores ou iguais a 0 ($\alpha \leq 0$), e caso as restrições sejam do tipo $Dx = d$, os multiplicadores devem ser irrestritos (α irrestrito).

Para encontrar o melhor limitante superior para o problema original P , ou seja, o menor valor possível para o problema relaxado $L_\alpha P$, deve-se encontrar um conjunto de valores “ótimos” para os multiplicadores lagrangianos α . Para isso, deve-se resolver um outro problema, conhecido como *dual lagrangiano* ($DL_\alpha P$). Esse problema é apresentado a seguir.

$$DL_\alpha P: \quad v(DL_\alpha P) = \text{Minimizar:} \quad v(L_\alpha P) \quad (2.31)$$

$$\text{Sujeito a:} \quad \alpha \geq 0 \quad (2.32)$$

A Relaxação Lagrangiana é apresentada de forma detalhada em Guignard (2003) e Wolsey (1998).

2.8.2 Relaxação Lagrangiana com *Clusters*

Uma outra estratégia usada para relaxar o problema original é a sua divisão em problemas menores e com as mesmas características. A divisão pode ser realizada pelo particionamento do grafo que representa o problema em *clusters* formados por vértices e arestas. Essa estratégia não garante a obtenção de uma solução viável para o problema completo, pois algumas arestas são ignoradas. Entretanto, uma maneira de considerar essas arestas é relaxá-las no sentido lagrangiano e encontrar um limitante de boa qualidade para o problema original. Assim, tem-se a idéia da “Relaxação Lagrangiana com *Clusters*” (LagClus) proposta por Ribeiro (2007), que é uma técnica recente e vem apresentando excelentes resultados para diferentes problemas de otimização (MAURI; LORENA, 2008b; RIBEIRO; LORENA, 2007; RIBEIRO; LORENA, 2008b).

Ribeiro (2007) considera a utilização de um *grafo de conflitos* (ATAMTÜRK et al., 2000) para representação do problema. Esse grafo estabelece uma relação lógica entre as variáveis binárias do modelo, ou seja, existe um vértice para cada variável, e uma aresta entre dois vértices quando no máximo uma das variáveis (vértices) puder ser igual a um na solução do problema.

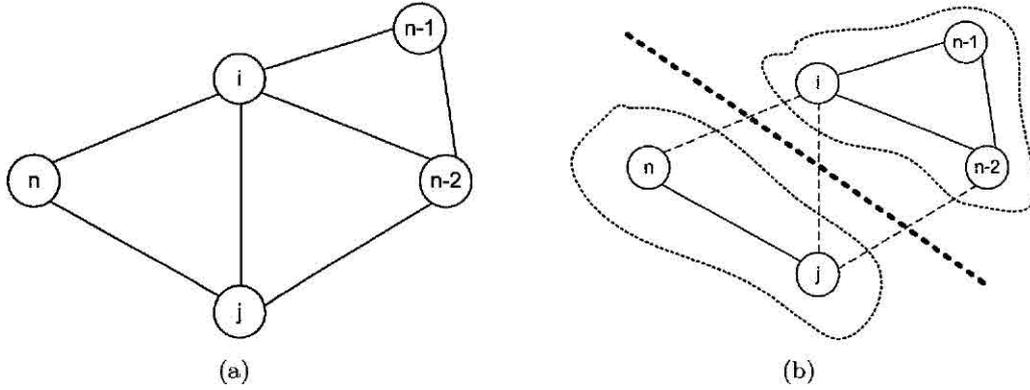


Figura 2.4 - Particionamento de um grafo na LagClus.

O primeiro passo para implementação da LagClus é a formação do grafo do problema. Considerando então um problema \bar{P} , pode-se definir um grafo $G = (V, A)$ baseado na restrição 2.34 (ver Figura 2.4).

\bar{P} : $v(\bar{P}) = \text{Maximizar:}$

$$\sum_{j=1}^n x_j \quad (2.33)$$

Sujeito a:

$$x_i + x_j \leq 1 \quad (i, j) \in A; \quad i, j = 1, \dots, n \quad (2.34)$$

$$x_j \in \{0, 1\} \quad j = 1, \dots, n \quad (2.35)$$

Nesse grafo, tem-se que $G = (V, A)$, com $V = \{1, \dots, n\}$ e a restrição 2.34 $\forall (i, j) \in A$, que indica a existência do conjunto de arestas A . É interessante destacar que, para o problema \bar{P} , o grafo será não direcionado. O grafo G pode ser particionado em p ($p \leq n$) subproblemas (*clusters*). Logo, tem-se $V = V_1 \cup V_2 \cup \dots \cup V_p$, onde $V_i \cap V_j = \emptyset$ ($\forall i, j = 1, \dots, p, i \neq j$), e $G_i = (V_i, A_i)$, para $i = 1, \dots, p$, $A_i = \{(k, l) \in A \text{ e } k \in V_i, l \in V_i\}$, e $X_i = V - V_i$, para $i = 1, \dots, p$. Assim, para o problema \bar{P} , tem-se p subproblemas ($k = 1, \dots, p$) formulados de acordo com o modelo a seguir.

$LC_\alpha \bar{P}_k$: $v(LC_\alpha \bar{P}_k) = \text{Maximizar:}$

$$\sum_{i \in V_k} x_i - \sum_{\substack{i \in V_k; \\ j \in X_k}} \alpha_{ij} x_i - \sum_{\substack{i \in X_k; \\ j \in V_k}} \alpha_{ij} x_j \quad (2.36)$$

Sujeito a:

$$x_i + x_j \leq 1 \quad (i, j) \in A; \quad i \in V_k \quad e \quad j \in V_k \quad (2.37)$$

$$x_j \in \{0, 1\} \quad j \in V_k \quad (2.38)$$

A solução da relaxação do problema \bar{P} em p clusters é dada pela Equação 2.39, e o dual lagrangiano correspondente pela Equação 2.40.

$$LC_\alpha \bar{P}^n : \quad v(LC_\alpha \bar{P}^n) = \sum_{k=1}^p v(LC_\alpha \bar{P}_k) + \sum_{\substack{(i,j) \in A; \\ cl(i) \neq cl(j)}} \alpha_{i,j} \quad (2.39)$$

$$DLC_\alpha \bar{P}^n : \quad v(DLC_\alpha \bar{P}^n) = \text{Min} \{v(LC_\alpha \bar{P}^n)\}, \alpha \geq 0 \quad (2.40)$$

No exemplo anterior (Figura 2.4), três arestas (tracejadas) deverão ser relaxadas no sentido lagrangiano (arestas de ligação $\Rightarrow cl(i) \neq cl(j)$: vértices i e j em clusters distintos), e as demais tratadas normalmente dentro dos subproblemas. Vale destacar que para esse problema \bar{P} , cada aresta representa diretamente uma restrição, logo, a relaxação no sentido lagrangiano é dada de forma análoga à apresentada na seção anterior, ou seja, essas restrições serão as consideradas restrições “difíceis”. A principal diferença da LagClus em relação à Relaxação Lagrangiana tradicional é que os subproblemas poderão ser resolvidos separadamente, ou seja, serão problemas menores, e possivelmente sua resolução será menos árdua.

Como descrito em Ribeiro (2007), os passos básicos da LagClus são:

- Dividir o grafo G em p clusters. O problema \bar{P} (sua função objetivo e restrições) agora é dividido em p subproblemas: $\bar{P}_1, \bar{P}_2, \dots, \bar{P}_p$.
- Relaxar as restrições de ligação no sentido lagrangiano, e deixar as demais nos respectivos subproblemas.
- Resolver os p subproblemas, e conseqüentemente a Relaxação Lagrangiana resultante para os multiplicadores α , que deverão ser atualizados para resolver o dual lagrangiano (Equação 2.40).

Por fim, Ribeiro (2007) ressalta que a resolução dos subproblemas gerados poderá ser tão difícil quanto o problema original. Nesse caso, pode-se optar por refazer

o particionamento com um número maior de *clusters*. A LagClus pode ser uma relaxação mais ou menos forte dependendo do número de *clusters* p . Quanto menor for p , mais forte é a relaxação, e quanto maior for p , mais próxima a LagClus estará da Relaxação Lagrangiana tradicional de todas as arestas em G , pois se $p = |V|$, cada vértice formará um *cluster*, e conseqüentemente todas as arestas em G serão relaxadas.

2.8.3 Decomposição Lagrangiana

A Decomposição Lagrangiana (DecLag) é um caso especial de Relaxação Lagrangiana que consiste em dividir o problema original em vários subproblemas e criar uma cópia das variáveis de decisão em cada um dos subproblemas gerados. Essas “cópias” são utilizadas nas restrições dos subproblemas, e restrições que garantem a igualdade entre as variáveis originais e suas cópias são relaxadas no sentido lagrangiano (CHARDAIRE; SUTTER, 1995).

Ao contrário da LagClus (ver Subseção 2.8.2), a DecLag independe da representação do problema por um grafo, entretanto, o seu uso facilita a compreensão dessa técnica. Considerando o mesmo problema \bar{P} utilizado na seção anterior, e o seu respectivo grafo (Figura 2.4), pode-se obter o grafo apresentado na Figura 2.5.

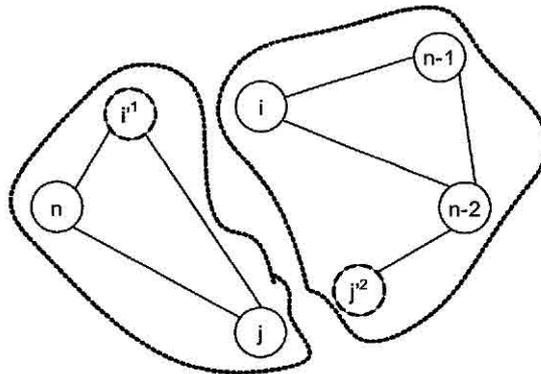


Figura 2.5 - Decomposição do problema \bar{P} .

Na Figura 2.5 se observa que dois vértices foram “copiados” (i^1 e j^2 são cópias dos vértices i e j , respectivamente), e conseqüentemente os dois subproblemas ficaram independentes (sem arestas de ligação). Entretanto, para obtenção de uma solução viável para o problema original \bar{P} , deve-se garantir que o valor das variáveis de

decisão referentes a esses vértices sejam iguais ao valor das variáveis originais correspondentes, e para isso, tem-se: $x_i = x_{j'}^1$ e $x_j = x_{j'}^2$ ($x_{j'}^2$ indica a cópia do vértice j no *cluster* 2, e $x_{j'}^1$ a cópia do vértice i no *cluster* 1) se $(i, j) \in A$ e $cl(i) \neq cl(j)$. Então, para que os subproblemas fiquem independentes, essas restrições devem ser relaxadas no sentido lagrangiano, e o dual lagrangiano correspondente otimizado. Logo, para o problema \bar{P} , tem-se p ($p \leq n$) subproblemas ($k = 1, \dots, p$) formulados de acordo com o modelo apresentado a seguir.

$DC_\alpha \bar{P}_k$: $v(DC_\alpha \bar{P}_k) = \text{Maximizar:}$

$$\sum_{i \in V_k} \left(- \sum_{d \neq k} \alpha_i^d \right) x_i + \sum_{\substack{\exists \alpha_{ij}=1; \\ i \in V_k; j \in X_k}} \alpha_j^k x_{j'}^k \quad (2.41)$$

Sujeito a:

$$x_i + x_j \leq 1 \quad (i, j) \in A; \quad i \in V_k \quad e \quad j \in V_k \quad (2.42)$$

$$x_i + x_{j'}^k \leq 1 \quad (i, j) \in A; \quad i \in V_k \quad e \quad j \in X_k \quad (2.43)$$

$$x_{j'}^k \in \{0, 1\} \quad j \in X_k \quad (2.44)$$

$$x_i \in \{0, 1\} \quad i \in V_k \quad (2.45)$$

A decomposição do problema \bar{P} em p *clusters* é dada pela Equação 2.46, e o dual lagrangiano correspondente pela Equação 2.47.

$$DC_\alpha \bar{P}^n : \quad v(DC_\alpha \bar{P}^n) = \sum_{k=1}^p v(DC_\alpha \bar{P}_k) \quad (2.46)$$

$$DDC_\alpha \bar{P}^n : \quad v(DDC_\alpha \bar{P}^n) = \text{Min} \{v(DC_\alpha \bar{P}^n)\}, \alpha \text{ irrestrito} \quad (2.47)$$

Mais detalhes sobre a DecLag são apresentados em Guignard (2003), e algumas aplicações são descritas em Billionnet et al. (1999), Chardaire e Sutter (1995) e Mauri e Lorena (2008a).

2.9 Algoritmo de Subgradientes

A resolução do dual lagrangiano $DL_\alpha P$ (ver Subseção 2.8.1), incluindo a escolha do valores apropriados para os multiplicadores α , não é uma tarefa simples, pois

o problema relaxado $L_\alpha P$ deverá ser resolvido para cada valor possível de α . Entretanto, $DL_\alpha P$ é convexo e linear por partes. Logo, pode-se utilizar o conceito de subgradientes para aproximar a solução do problema relaxado à solução ótima do problema original (PARKER; RARDIN, 1988). O primeiro, e mais utilizado, algoritmo para resolver essa relaxação é o Algoritmo de Subgradientes (ver Figura 2.6).

- | |
|--|
| <ol style="list-style-type: none"> 1. <u>INICIALIZAR</u> (o vetor de multiplicadores α^0 ($\alpha \geq 0$)); 2. <u>CONSIDERAR</u> ($lb = -\infty$ e $ub = +\infty$ - limitantes inferiores e superiores); 3. <u>REPETIR</u> 4. <u>RESOLVER</u> (o modelo relaxado $L_\alpha P$ obtendo um vetor de soluções x_α e $v(L_\alpha P)$; 5. <u>OBTER</u> (uma solução viável para P e o correspondente valor $v(P)$); 6. <u>ATUALIZAR</u> ($lb = \max[lb, v(P)]$); 7. <u>ATUALIZAR</u> ($ub = \min[ub, v(L_\alpha P)]$); 8. <u>CALCULAR</u> (os subgradientes $\varphi = Dx_\alpha - d$); 9. <u>CALCULAR</u> (o passo ε: $\varepsilon = \frac{\eta(ub - lb)}{\ \varphi\ ^2}$); 10. <u>ATUALIZAR</u> (os multiplicadores α: $\alpha^{t+1} = \text{Max}(0, \alpha^t + \varepsilon_t(Dx - d))$); 11. <u>ATÉ</u> (que seja atingido um critério de parada); |
|--|

Figura 2.6 - Algoritmo de Subgradientes.

A Figura 2.6 apresenta o Algoritmo de Subgradientes baseado nas definições apresentadas em Narciso e Lorena (1999). Nesse algoritmo, os multiplicadores lagrangianos α são inicializados com valores quaisquer ($\alpha \geq 0$ para o problema P). Logo, o modelo relaxado $L_\alpha P$ é resolvido, e sua solução (x_α) é utilizada para obter uma solução viável para o problema original P e para atualizar os multiplicadores α e os subgradientes φ . Uma solução viável para o problema original P pode ser obtida por diferentes métodos, sendo estes chamados de “heurística primal” ou “heurística lagrangiana”. Outra forma de considerar esse algoritmo é utilizar o valor de uma solução viável já conhecida (que não se sabe se é ótima) como limitante inferior (em problemas de maximização) durante todo o processo. Nesse caso, a idéia é verificar a “qualidade” dessa solução já conhecida, e conseqüentemente não é necessário que se busque uma solução viável para o problema original a cada iteração.

Os subgradientes são atualizados de acordo com as restrições relaxadas, e a atualização dos multiplicadores ainda utiliza um conceito de “passo” (ε), que direciona a busca pelo melhor conjunto de valores para os multiplicadores. Para a definição desse passo, ainda é utilizado um parâmetro de controle η , que foi proposto por Held e Karp (1970).

Geralmente, como definido em Held e Karp (1970), o valor inicial de η é 2, e este é dividido pela metade caso o valor da solução relaxada - $v(L_\alpha P)$ - não diminua por 30 iterações sucessivas, por exemplo. Como critério de parada, pode-se considerar um número máximo de iterações, um tempo limite de processamento, e um *gap* limite entre os valores das soluções primal e dual ($ub = lb$).

3 PROBLEMA *DIAL-A-RIDE*

Dentre a classificação dos problemas de roteirização e programação de veículos, aqueles que envolvem o transporte de passageiros de seus locais de origem a seus destinos são conhecidos genericamente na literatura como problemas do tipo “*dial-a-ride*” (ZNAMENSKY; CUNHA, 1999).

Como provado por Baugh et al. (1998), esses problemas são classificados como *NP-Hard*, ou seja, não se conhece um algoritmo com tempo polinomial em função do tamanho da entrada (número de requisições de transporte e veículos disponíveis) capaz de, dada uma solução, verificar se esta é a de custo mínimo (solução ótima).

O *Problema de Roteirização e Programação de Veículos*, ou *Dial-a-Ride Problem* (DARP) (CORDEAU; LAPORTE, 2007), consiste no desenvolvimento de rotas e escalas de veículos para transportar diversos usuários, os quais especificam requisições de embarque e desembarque (coleta e entrega) entre locais de origem e destino específicos. O objetivo desse processo é planejar um conjunto de rotas (com custo mínimo) para alguns veículos que seja capaz de acomodar o maior número possível de usuários, sempre obedecendo a um conjunto de restrições.

Em se tratando especificamente do caso de transporte de *Pessoas Portadoras de Necessidades Especiais* (PPNEs), um ponto de embarque (coleta) corresponde ao endereço onde uma determinada PPNE deve ser buscada, e o ponto de desembarque (entrega) associado corresponde ao endereço onde a mesma PPNE deve ser entregue. Cada ponto de embarque e seu respectivo ponto de desembarque, juntamente com suas respectivas “janelas de tempo”, formam uma *Requisição de Transporte* de um cliente.

Uma tendência comum nos modelos do DARP é deixar que os usuários determinem uma “janela de tempo” (isto é, janelas ou intervalos de horários para seu atendimento) para sua partida e sua chegada, pois segundo Jaw et al. (1986), os usuários devem estar aptos a especificar um intervalo de horários para seu embarque e desembarque, ambos em locais específicos, possibilitando assim um melhor planejamento da programação e roteirização dos veículos.

Cada veículo possui uma capacidade, medida normalmente em número de assentos convencionais e número de cadeiras de rodas, por exemplo. Analogamente, a cada solicitação de atendimento está associada uma ocupação em termos de quantidade

de assentos convencionais (eventuais acompanhantes) e quantidade de cadeiras de rodas (ZNAMENSKY; CUNHA, 1999).

Segundo Cordeau e Laporte (2003b), Cordeau e Laporte (2007) e outros diversos autores, o DARP pode ser abordado de forma estática ou dinâmica. No primeiro caso, todas as requisições de transporte são conhecidas a priori, enquanto no segundo caso são consideradas requisições feitas ao longo do dia (normalmente por telefone), e as rotas dos veículos são ajustadas em tempo real de acordo com a demanda. Porém, na prática, raramente existem DARPs dinâmicos “puros”, pois normalmente um subconjunto de requisições é conhecido com antecedência.

A maioria dos estudos acerca do DARP assume a disponibilidade de um conjunto de veículos homogêneos instalados em uma única garagem, entretanto, é importante perceber que na prática existem situações diferentes, como por exemplo: pode haver várias garagens, especialmente em grandes áreas geográficas, e o conjunto de veículos às vezes é heterogêneo, sendo alguns veículos projetados para só transportar cadeiras de rodas, outros só para transportar passageiros para ambulatorios, e ainda alguns capazes de acomodar ambos os tipos de passageiros (CORDEAU; LAPORTE, 2003b; CORDEAU; LAPORTE, 2007).

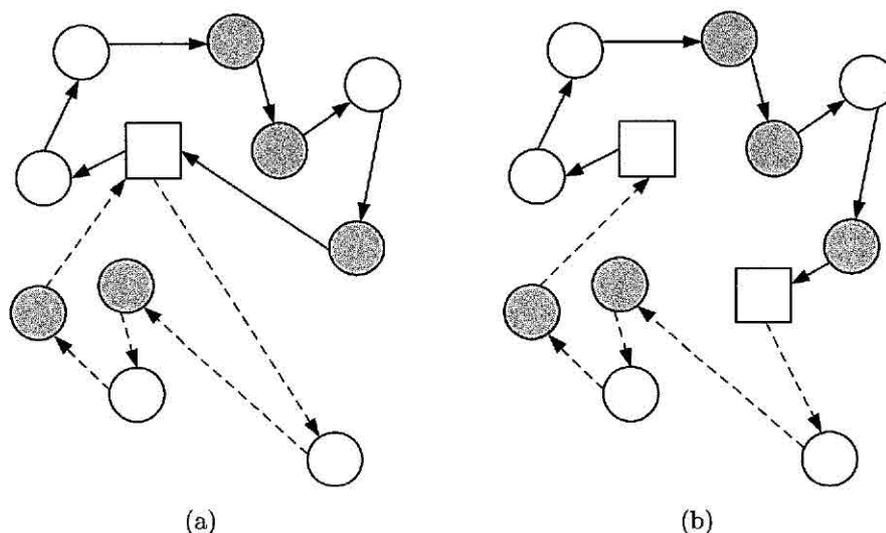


Figura 3.1 - Casos distintos do DARP.

A Figura 3.1 apresenta um exemplo de solução para dois casos distintos do DARP: garagem única (a), e garagens múltiplas (b). Nessa figura, os círculos brancos indicam

os locais de embarque, e os cinzas indicam os locais de desembarque. As garagens são representadas pelos quadrados, e nesse caso, tem-se duas rotas (dois veículos), uma representada pelas linhas contínuas e outra pelas tracejadas.

O DARP pode ser considerado com vários objetivos distintos, como por exemplo: minimizar os custos operacionais sujeito à satisfação de toda a demanda; maximizar a satisfação da demanda, sujeito à disponibilidade dos veículos, ou outras combinações destes, que geralmente buscam um equilíbrio entre custo operacional e qualidade do serviço prestado (PAQUETTE et al., 2008).

Normalmente, o critério de avaliação utilizado na resolução do DARP inclui a distância total percorrida pelos veículos, a duração das rotas, o tempo médio de espera dos veículos nos locais de embarque e desembarque, e o tempo médio de viagem dos clientes (i.e., tempo de permanência dos clientes dentro dos veículos), sendo dado a cada um desses, um maior ou menor grau de importância em relação aos outros.

Em termos gerais, o DARP engloba vários problemas de roteamento de veículos como o *Pick-up and Delivery Vehicle Routing Problem* (PDVRP) e o *Vehicle Routing Problem with Time Windows* (VRPTW). O que faz o DARP diferente da maioria de tais problemas é a perspectiva humana, ou seja, no transporte de passageiros, a redução da “inconveniência” dos usuários (qualidade do serviço) deve ser equilibrada com a minimização dos custos operacionais (CALVO et al., 2004; CORDEAU; LAPORTE, 2003b).

3.1 Revisão da literatura

O problema *dial-a-ride* (DARP) é muito comum, porém relativamente pouco difundido dentro da Pesquisa Operacional e da Otimização Combinatória devido a sua alta complexidade. Entretanto, alguns pesquisadores vêm o estudando, e consequentemente apresentando métodos variados de solução e trabalhos realizados segundo seus preceitos e aplicações.

3.1.1 Abordagens clássicas

Psaraftis (1980) desenvolveu um algoritmo exato de *Programação Dinâmica* para a resolução de um problema de coleta e entrega para um único veículo, sem restrições de janelas de tempo, considerando os casos estático e dinâmico do problema. A função objetivo adotada foi a minimização de uma ponderação entre o tempo total

de operação do veículo e a “insatisfação” total dos usuários, a qual é medida por uma função linear dos tempos de espera e viagem de cada usuário. A extensão do problema para a inclusão de restrições de janelas de tempo também foi analisada por Psaraftis (1983), e uma melhoria na solução obtida nessa extensão, também por meio de programação dinâmica, foi publicada por Desrosiers et al. (1986).

O trabalho proposto por Jaw et al. (1986) trata o problema de uma forma mais próxima a sua essência, ou seja, com múltiplos veículos e com restrições de janelas de tempo. Eles apresentam uma heurística denominada “*Heurística de Inserção Paralela*”, cuja idéia central consiste em ordenar as solicitações de atendimento por um critério de horário, e inserí-las, segundo a ordenação adotada, nas rotas que resultem em menor acréscimo de custo. Essa heurística foi testada com sucesso para uma base de dados de aproximadamente 2600 clientes e 20 veículos. É interessante destacar que essas instâncias representavam o problema para vários dias de operação, ou seja, nem todos os clientes solicitavam atendimento no mesmo dia.

Dumas et al. (1991) desenvolveram um algoritmo exato, baseado no método de *Geração de Colunas*, para o problema com frota homogênea e restrições de janelas de tempo. Segundo os autores, o algoritmo foi desenvolvido tendo em vista instâncias do problema nas quais as restrições de capacidade dos veículos fossem fortemente restritivas e atuantes. O algoritmo foi aplicado com sucesso a problemas com até 55 clientes.

Uma modificação da *Heurística de Inserção Paralela* de Jaw et al. (1986) foi desenvolvida por Madsen et al. (1995). A diferença central é a adoção de um critério de “dificuldade” para ordenação das solicitações. Este critério procura priorizar a inserção de solicitações com restrições mais rígidas, tornando a inserção mais eficiente, pois nos estágios iniciais da heurística, os veículos nas rotas estão mais vazios, e há um número maior de alternativas de inserções viáveis.

Toth e Vigo (1996) descrevem uma *Heurística de Inserção Paralela* para o problema com múltiplos veículos e restrições de janelas de tempo. Também é apresentado um método de melhoria da solução obtida por meio de procedimentos de busca local. Tais procedimentos são divididos em duas categorias: trocas intra-rotas e trocas inter-rotas. A primeira categoria corresponde a tentativas de melhoria das rotas pela modificação da sequência dos locais visitados dentro de cada rota. Já a segunda categoria considera a possibilidade de troca de requisições de transporte entre rotas

distintas. A heurística foi testada com sucesso pelos autores em um conjunto de cerca de 1500 solicitações provenientes de uma situação real. Um aprimoramento desse método de melhoria da solução foi desenvolvido por Toth e Vigo (1997), com a inclusão de um método de busca local derivado da busca tabu. Esse refinamento provou ser capaz de melhorar os resultados obtidos pela versão original do processo de melhoria.

Baugh et al. (1998) utilizam o *Simulated Annealing* para resolver o DARP. Nesse trabalho, é utilizada uma abordagem de “agrupar-primeiro” e “rotear-depois”. Nessa abordagem, os clientes são inicialmente divididos em grupos (*clusters*), e depois para cada grupo é determinada a ordem de atendimento de seus clientes. O agrupamento é feito pelo *Simulated Annealing* e o roteamento por uma heurística do vizinho mais próximo. Os resultados foram obtidos para instâncias com até 300 clientes, e os tempos de processamento para obtenção desses resultados não são apresentados no artigo.

Znamensky e Cunha (1999) tratam o DARP por meio de uma adaptação da *Heurística de Inserção Paralela* proposta por Madsen et al. (1995), seguida de uma etapa de melhoria das rotas obtidas similar à apresentada por Toth e Vigo (1996). Nesse trabalho, é tratado especificamente o atendimento de idosos e deficientes por veículos de pequena capacidade (peruas ou vans), e as instâncias utilizadas representam um caso real extraído da operação do ATENDE, serviço de transporte de deficientes gerido pela São Paulo Transporte S/A. Essas instâncias correspondem a um dia típico do mês de novembro de 1998, e perfazem 349 requisições de transporte (clientes) e 84 veículos disponíveis, localizados em 47 garagens distribuídas no município de São Paulo.

3.1.2 Abordagens recentes

Jih et al. (2002) utilizam um *Algoritmo Genético* para resolver um problema de coleta e entrega com janelas de tempo e apenas um veículo. Nesse algoritmo, a representação de uma rota é definida por um cromossomo que representa a sequência de locais a serem atendidos pelo veículo. O algoritmo ainda permite a exploração de soluções inválidas durante a busca. A função objetivo trata apenas os custos de viagem do veículo e as violações nas restrições. Nesse trabalho, são utilizadas instâncias geradas aleatoriamente com até 100 clientes, sendo o tempo de processamento necessário para resolvê-las de aproximadamente 38 minutos.

Cordeau e Laporte (2003a) apresentam uma abordagem que utiliza a metaheurística *Busca Tabu* para resolver o DARP. Nessa abordagem o problema é tratado na sua forma estática, com múltiplos veículos, frota homogênea e garagem única. Além disso, são comparados três métodos heurísticos, *P1*, *P2* e *P3*, que, juntamente com a *Busca Tabu*, são utilizados para resolver o problema. O primeiro (*P1*) visa apenas à minimização das violações nas janelas de tempo, enquanto o segundo (*P2*), além de tentar minimizar essas violações, visa também à minimização da duração rotas. Já o terceiro (*P3*) tem esses mesmos objetivos, porém visando também a redução nos tempos de viagem dos clientes. Os resultados obtidos mostram que *P1* e *P2*, por serem mais “simples”, são mais rápidos que *P3*, porém as soluções obtidas por *P3* são expressivamente melhores. Todos os resultados são obtidos a partir de instâncias geradas aleatoriamente baseadas em informações cedidas pela *Montreal Transit Commission* (MTC) de Montreal - Canadá. Essas instâncias variam de 24 a 144 clientes. Além disso, eles apresentam outros resultados obtidos para instâncias de problemas reais (com 200 e 295 clientes) cedidas por uma transportadora dinamarquesa.

Jorgensen et al. (2007) apresentam uma abordagem que utiliza um *Algoritmo Genético* para resolver o DARP. Nessa abordagem o problema é tratado na sua forma estática, com múltiplos veículos, frota heterogênea e garagens múltiplas. A resolução do problema é obtida por uma abordagem clássica de “agrupar primeiro” e “rotear depois”. O Algoritmo Genético é utilizado para agrupar (distribuir) os clientes nos veículos, ou seja, determinar quais serão os clientes atendidos por cada veículo. Já o roteamento (sequência de atendimento) e a programação (determinação dos horários) são determinados, independentemente para cada veículo, por uma heurística específica. Os resultados são obtidos a partir das instâncias geradas aleatoriamente por Cordeau e Laporte (2003a).

Cordeau (2006) apresenta uma abordagem que utiliza o algoritmo *Branch-and-Cut* para resolver o DARP. Nessa abordagem o problema é tratado na sua forma estática, com múltiplos veículos, frota homogênea e garagem única. A resolução do problema é realizada de forma exata, garantindo assim a obtenção da solução ótima para o problema. Os resultados são obtidos a partir de instâncias geradas aleatoriamente com no máximo 48 clientes.

Algumas heurísticas “simples” e específicas para o DARP são apresentadas em outros diversos trabalhos recentes (WONG; BELL, 2006; XIANG et al., 2006). Além disso,

alguns trabalhos apresentam sistemas computacionais baseados em heurísticas e metaheurísticas para solucionar casos reais do DARP (CALVO et al., 2004; MELACHINOUDIS et al., 2007; REKIEK et al., 2006), e Paquette et al. (2008) apresentam uma discussão sobre os critérios de qualidade a serem analisados na solução deste problema. Por fim, algumas revisões mais abrangentes e profundas sobre o DARP são apresentadas por Cordeau e Laporte (2003b), Cordeau et al. (2006) e Cordeau e Laporte (2007).

3.2 Modelos existentes

Nesta seção são apresentados dois modelos matemáticos utilizados para representar o DARP. Esses modelos são bem conhecidos, e foram apresentados inicialmente em trabalhos de relevante importância para o estudo do DARP.

Assume-se inicialmente a existência de n clientes (requisições de transporte) a serem atendidos por m veículos. Cada requisição de transporte especifica um local de embarque i e um de desembarque $n+i$. Para representar o problema então, são definidos os seguintes conjuntos:

- K : conjunto dos veículos disponíveis ($|K| = m$).
- G^- : conjunto de garagens de origem.
- G^+ : conjunto de garagens de destino.
- P : conjunto dos locais de embarque.
- U : conjunto dos locais de desembarque.
- $N = \{G^- \cup P \cup U \cup G^+\}$: conjunto com todos os locais (pontos).

Cada cliente i ($\forall i \in P$) especifica a “carga” q_i necessária para seu transporte, ou seja, o número de assentos no veículo que ele irá ocupar, e também dois intervalos de horários, um em que ele gostaria de embarcar na sua origem $[e_i, l_i]$, e outro que ele gostaria de desembarcar no seu destino $[e_{n+i}, l_{n+i}]$ (janelas de tempo). A carga q_i deverá ser um valor positivo nos locais de embarque e o mesmo valor, porém negativo, nos respectivos locais de desembarque.

Cada veículo k ($\forall k \in K$) tem uma capacidade conhecida Q^k (quantidade de assentos disponíveis), um tempo máximo de duração associado a sua rota T^k , e começa

sua rota em uma garagem específica g_k^- e termina em uma outra g_k^+ , podendo ser a mesma ou não. Todas as garagens ainda têm suas próprias janelas de tempo para partida (garagem de origem) e chegada (garagem de destino).

A cada cliente i ($\forall i \in P$) é associado um tempo máximo de viagem R^i , ou seja, o tempo máximo em que o cliente poderá ficar dentro do veículo. A cada local i ($\forall i \in \{P \cup U\}$) é associado um tempo máximo de espera W^i , ou seja, o tempo máximo em que os veículos poderão ficar esperando até iniciar o “serviço”. Considera-se como serviço o embarque ou desembarque de um cliente em um determinado local, sendo atribuído um tempo s_i necessário para sua completude. Por fim, dados os locais referentes às garagens e aos pontos pertencentes às requisições, têm-se as distâncias $d_{i,j}$, as durações das viagens $t_{i,j}$ e os custos de deslocamento $c_{i,j}$ entre os pontos i e j , $i, j \in N$ e $i \neq j$.

Existe ainda um conjunto de requisitos essenciais que devem ser obedecidos, ou seja, requisitos que obrigatoriamente devem ser atendidos para que se obtenha uma solução válida (viável em termos práticos) para o problema. Dentre esses requisitos estão:

- A duração da rota executada pelo veículo k ($\forall k \in K$) não deve exceder o tempo máximo permitido T^k .
- O tempo de viagem do cliente i ($\forall i \in P$) não deve exceder o tempo máximo de viagem permitido R^i .
- O tempo de espera no local i ($\forall i \in \{P \cup U\}$) não deve exceder o tempo máximo de espera permitido W^i .
- A capacidade Q^k ($\forall k \in K$) dos veículos não pode ser excedida em nenhum local.
- O início do serviço em todos os locais i ($\forall i \in N$) deve estar dentro dos intervalos pré-estabelecidos $[e_i, l_i]$.

O objetivo então é minimizar os custos operacionais e/ou a “insatisfação” dos clientes, ou seja, minimizar os requisitos “não-essenciais” do problema, sempre atendendo aos requisitos essenciais.

De posse das informações, pode-se definir as variáveis de decisão que irão fornecer toda a roteirização e a programação dos veículos. Considera-se A_i^k como sendo o horário de chegada no local i ($\forall i \in N$) pelo veículo k , sendo $A_i^k = 0$ se $i \in G^-$ e $A_i^k = D_{i-1}^k + t_{i-1,i}$ se $i \in \{P \cup U \cup G^+\}$; D_i^k indica o horário de partida do local i ($\forall i \in N$), novamente pelo veículo k , sendo $D_i^k = 0$ se $i \in G^+$ e $D_i^k = B_i^k + s_i$ se $i \in \{P \cup U\}$ e $D_i^k = B_i^k$ se $i \in G^-$; B_i^k representa o horário de início do serviço no local i ($\forall i \in N$), sendo $B_i^k = D_i^k$ se $i \in G^-$ e $B_i^k = \max\{e_i, A_i^k\}$ se $i \in \{P \cup U \cup G^+\}$; o tempo de espera antes do início do serviço no local i ($\forall i \in N$) é dado por W_i^k , sendo $W_i^k = 0$ se $i \in G^-$ e $W_i^k = B_i^k - A_i^k$ se $i \in \{P \cup U \cup G^+\}$; Q_i^k indica a carga (número de assentos ocupados) do veículo k após o término do serviço no local i ($\forall i \in N$), sendo $Q_i^k = 0$ se $i \in \{G^- \cup G^+\}$ e $Q_i^k = Q_{i-1}^k + q_i$ se $i \in \{P \cup U\}$; e por fim, considera-se R_i^k como sendo o tempo de viagem do cliente i ($\forall i \in P$) no veículo k , sendo $R_i^k = B_{n+i}^k - D_i^k$.

A partir de então, considera-se $x_{i,j}^k = 1$ caso o veículo k se desloque do local i para o local j e $x_{i,j}^k = 0$ caso contrário.

3.2.1 Modelo proposto por Cordeau (2006)

A abordagem apresentada por Cordeau (2006) trata o DARP de forma estática (todas as requisições são conhecidas a priori), com múltiplos veículos, com frota homogênea (todos os veículos possuem uma mesma capacidade) e garagem única (todos os veículos começam e terminam sua rota na mesma garagem). O modelo é apresentado a seguir:

Minimizar:

$$\sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{i,j} x_{i,j}^k \quad (3.1)$$

Sujeito a:

$$\sum_{k \in K} \sum_{j \in N} x_{i,j}^k = 1 \quad \forall i \in P \quad (3.2)$$

$$\sum_{j \in N} x_{i,j}^k - \sum_{j \in N} x_{n+i,j}^k = 0 \quad \forall k \in K; i \in P \quad (3.3)$$

$$\sum_{j \in N} x_{g_k^-,j}^k = 1 \quad \forall k \in K \quad (3.4)$$

$$\sum_{j \in N} x_{j,i}^k - \sum_{j \in N} x_{i,j}^k = 0 \quad \forall k \in K; i \in \{P \cup D\} \quad (3.5)$$

$$\sum_{i \in N} x_{i, g_k^+}^k = 1 \quad \forall k \in K \quad (3.6)$$

$$B_j^k \geq (B_i^k + s_i + t_{ij}^k) x_{i,j}^k \quad \forall k \in K; i, j \in N \quad (3.7)$$

$$Q_j^k \geq (Q_i^k + q_j) x_{i,j}^k \quad \forall k \in K; i, j \in N \quad (3.8)$$

$$R_i = B_{n+i}^k - (B_i^k + s_i) \quad \forall k \in K; i \in P \quad (3.9)$$

$$B_{g_k^+}^k - B_{g_k^-}^k \leq T_k \quad \forall k \in K \quad (3.10)$$

$$e_i \leq B_i^k \leq l_i \quad \forall k \in K; i \in N \quad (3.11)$$

$$t_{i, n+i} \leq R_i^k \leq R_i \quad \forall k \in K; i \in P \quad (3.12)$$

$$\max(0, q_i) \leq Q_i^k \leq \min(Q_k, Q_k + q_i) \quad \forall k \in K; i \in N \quad (3.13)$$

$$x_{i,j}^k \in \{0, 1\} \quad \forall k \in K; i, j \in N \quad (3.14)$$

A função objetivo 3.1 minimiza o custo total da rota, que pode ser obtido pela distância total percorrida pelos veículos ($c_{i,j} = d_{i,j}$), pelo tempo total percorrido pelos veículos ($c_{i,j} = t_{i,j}$), etc. As restrições 3.2 e 3.3 garantem que cada requisição será atendida por exatamente um único veículo, e que um local de embarque e seu respectivo local de desembarque sempre serão atendidos pelo mesmo veículo.

As restrições 3.4, 3.5 e 3.6 garantem que a rota de cada veículo k irá começar e terminar na garagem. Nessas restrições, $g_k^- = g_k^+$, ou seja, só existe uma garagem. A consistência nos cálculos dos horários e da carga dos veículos é garantida nas restrições 3.7 e 3.8. A restrição 3.9 define o tempo de viagem de cada cliente, que é “limitado” pela restrição 3.12. Já a restrição 3.10 limita o tempo de duração de cada rota, enquanto as restrições 3.11 e 3.13 garantem, respectivamente, a não violação das janelas de tempo e da capacidade dos veículos. Por fim, a restrição 3.14 apenas garante que as variáveis de decisão $x_{i,j}^k$ sejam binárias.

Como pode ser observado, devido às restrições 3.7 e 3.8, esta formulação é não-linear. Entretanto, esse modelo pode ser linearizado pela introdução de constantes $M_{i,j}^k$ e $Y_{i,j}^k$, substituindo as restrições 3.7 e 3.8 pelas restrições 3.15 e 3.16.

$$B_j^k \geq B_i^k + s_i + t_{i,j} - M_{i,j}^k(1 - x_{i,j}^k) \quad \forall k \in K; i, j \in N \quad (3.15)$$

$$Q_j^k \geq Q_i^k + q_j - Y_{i,j}^k(1 - x_{i,j}^k) \quad \forall k \in K; i, j \in N \quad (3.16)$$

onde:

$$M_{ij}^k \geq \max\{0, l_i + s_i + t_{ij} - e_j\} \quad \text{e} \quad Y_{ij}^k \geq \min\{Q_k, Q_k + q_i\}$$

Além disso, Cordeau (2006) ainda propõe uma redução no número de variáveis e restrições desse modelo por meio da agregação das variáveis de tempo B_i^k em apenas B_i , ou seja, como apenas um veículo irá atender o local i , não é necessário que se tenha uma variável para cada veículo atendendo o local i , exceto na garagem, onde todos os veículos irão “passar”. Nesse caso, basta substituir as restrições 3.7 e 3.9 pelas seguintes:

$$B_j^k \geq (B_{g_k^-}^k + s_{g_k^-} + t_{g_k^-,j})x_{g_k^-,j}^k \quad \forall k \in K; j \in N \quad (3.17)$$

$$B_j \geq (B_i + s_i + t_{i,j}) \sum_{k \in K} x_{i,j}^k \quad \forall i, j \in N \quad (3.18)$$

$$B_{g_k^+}^k \geq (B_i + s_i + t_{i,j})x_{i,g_k^+}^k \quad \forall k \in K; i \in N \quad (3.19)$$

$$R_i = B_{n+i} - (B_i + s_i) \quad \forall i \in P \quad (3.20)$$

Cordeau (2006) também propõe a eliminação de alguns “arcos” do problema, como por exemplo, da garagem para um local de desembarque, de um local de desembarque para seu respectivo local de embarque, etc. Essa eliminação é detalhada na Seção 3.3.

Este modelo é interessante e apresenta características que “simplificam” a representação do problema, como por exemplo, a redução no número de variáveis e restrições. Entretanto, essa abordagem é restrita a casos que raramente irão condizer com a realidade, pois trata apenas problemas com frota homogênea, garagem única e com custo fixo, definido por uma função objetivo “simples”, impossibilitando uma representação que busque a obtenção de uma solução equilibrada entre os custos operacionais e a inconveniência dos clientes.

3.2.2 Modelo proposto por Bergvinsdottir (2004)

A abordagem apresentada por Bergvinsdottir (2004) também trata o DARP de forma estática, com múltiplos veículos, com frota heterogênea (cada veículo possui uma capacidade distinta) e garagens múltiplas (cada veículo começa e termina sua rota em garagens específicas). O modelo é apresentado a seguir:

Minimizar:

$$\omega_0 \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} t_{i,j} x_{i,j}^k + \omega_1 m + \omega_2 \sum_{k \in K} (B_{d(k)}^k - B_{o(k)}^k) + \quad (3.21)$$

$$\omega_3 \sum_{k \in K} \sum_{i \in P} (T_{n+i}^k - s_i - T_i^k - t_{i,n+i}) + \omega_4 \sum_{k \in K} \sum_{i \in \{PUU\}} W_i^k (Q_i^k - q_i) \quad (3.22)$$

Sujeito a:

$$\sum_{k \in K} \sum_{j \in PUg_k^+} x_{g_k^-, j}^k = m \quad (3.23)$$

$$\sum_{k \in K} \sum_{j \in DUg_k^-} x_{i, g_k^+}^k = m \quad (3.24)$$

$$\sum_{j \in N} x_{i,j}^k - \sum_{j \in N} x_{j,i}^k = 0 \quad \forall k \in K; i \in \{P \cup U\} \quad (3.25)$$

$$\sum_{k \in K} \sum_{j \in \{PUU\}} x_{i,j}^k = 1 \quad \forall i \in P \quad (3.26)$$

$$\sum_{j \in \{PUU\}} x_{i,j}^k - \sum_{j \in \{PUU\}} x_{j,n+i}^k = 0 \quad \forall k \in K; i \in P \quad (3.27)$$

$$x_{i,j}^k (B_i^k + s_i + t_{i,j} + W_j^k - B_j^k) \leq 0 \quad \forall k \in K; i, j \in N \quad (3.28)$$

$$B_i^k + s_i + t_{i,n+i} + W_j^k - B_{n+i}^k \leq 0 \quad \forall k \in K; i \in P \quad (3.29)$$

$$e_i \leq B_i^k \leq l_i \quad \forall k \in K; i \in \{P \cup U\} \quad (3.30)$$

$$x_{i,j}^k (Q_i^k + q_j - Q_j^k) = 0 \quad \forall k \in K; i, j \in N \quad (3.31)$$

$$q_i \leq Q_i^k \leq Q_k \quad \forall k \in K; i \in P \quad (3.32)$$

$$Q_{g_k^-}^k = Q_{g_k^+}^k = 0 \quad \forall k \in K \quad (3.33)$$

$$B_{g_k^+}^k - B_{g_k^-}^k \leq T_k \quad \forall k \in K \quad (3.34)$$

$$B_{n+i}^k + B_i^k \leq R_i \quad \forall k \in K; i \in P \quad (3.35)$$

$$x_{i,j}^k \in \{0, 1\} \quad \forall k \in K; i, j \in N \quad (3.36)$$

A função objetivo (3.21 e 3.22) é uma função multi-critério (multi-objetivo) que visa minimizar o custo total de transporte (3.21 - tempo de viagem entre os locais, número de veículos usados e duração das rotas) e a inconveniência dos clientes (3.22 - tempo total de viagem e tempo total de espera). O tratamento dessa função

multi-objetivo é dado pela multiplicação de cada termo por um vetor de penalização $\omega = [\omega_0, \omega_1, \omega_2, \omega_3, \omega_4]$. Esse vetor é usado então para determinar a importância de cada requisito no problema como um todo.

Segundo Bergvinsdottir (2004), as restrições 3.23 e 3.24 garantem que cada veículo comece e termine sua rota nas garagens específicas. A restrição 3.25 garante que todos os locais sejam visitados. As restrições 3.26 e 3.27 garantem que cada cliente irá embarcar e desembarcar no mesmo veículo. A restrição 3.28 assegura que o horário de chegada no local j seja maior do que a soma do horário de partida do local i mais o tempo de viagem entre i e j . Já a restrição 3.29 garante que um cliente irá embarcar antes de desembarcar, ou seja, o veículo que irá atendê-lo passará primeiro no local de embarque desse cliente, e só depois no seu local de desembarque. As janelas de tempo são respeitadas pela restrição 3.30. A consistência nos cálculos da carga dos veículos é garantida na restrição 3.31. A restrição 3.32 garante que a capacidade do veículo não será excedida em nenhum local. A restrição 3.33 indica que os veículos sempre estarão vazios nas garagens. A restrição 3.34 garante que a duração da rota não irá exceder o máximo permitido. A restrição 3.35 garante que o tempo de viagem dos clientes não irá exceder o tempo máximo permitido. Por fim, a restrição 3.36 apenas garante que as variáveis de decisão $x_{i,j}^k$ sejam binárias.

Assim como no modelo apresentado na Subseção 3.2.1, esta formulação também é não-linear, porém pode ser linearizada pela introdução de uma constante M , substituindo as restrições 3.28 e 3.31 pelas restrições 3.37, 3.38 e 3.39.

$$B_i^k + s_i + t_{i,j} + W_j^k - B_j^k - M(1 - x_{ij}^k) \leq 0 \quad \forall k \in K; i, j \in N \quad (3.37)$$

$$Q_i^k + q_j - Q_j^k - M(1 - x_{ij}^k) \leq 0 \quad \forall k \in K; i, j \in N \quad (3.38)$$

$$Q_i^k + q_j - Q_j^k + M(1 - x_{ij}^k) \geq 0 \quad \forall k \in K; i, j \in N \quad (3.39)$$

M é um valor muito grande (sempre maior do que $B_i^k + s_i + t_{i,j} + W_j^k - B_j^k$ e $Q_i^k + q_j - Q_j^k$).

Após linearizar as restrições 3.28 e 3.31, o modelo ainda continua não-linear, pois o último termo da função objetivo (que calcula o tempo total de espera) é não-linear, o que contradiz Bergvinsdottir (2004), que afirma que seu modelo é linear. Outro equívoco do autor é afirmar que as restrições 3.23 e 3.24 garantem que cada veículo comece e termine sua rota nas garagens. Isso também não ocorre, pois essas restrições

permitem, por exemplo, que um veículo saia e chegue mais de uma vez na garagem. Por fim, alguns prováveis “erros de digitação” são apresentados nas restrições 3.29 e 3.35. Na restrição 3.29 deveria ser W_{n+i}^k ao invés de W_j^k , e a restrição 3.35 deveria ser formada por uma subtração e não uma adição.

Ao contrário de Cordeau (2006), Bergvinsdottir (2004) não trata a redução no número de variáveis e restrições do modelo e a eliminação de arcos do problema. Entretanto, esse modelo é mais condizente com a realidade, pois trata o problema com frota heterogênea, garagens múltiplas e com uma função multi-objetivo, o que possibilita a busca por soluções equilibradas entre os custos operacionais e a inconveniência dos clientes.

3.3 Abordagem voltada para a qualidade do serviço

Apesar de algumas “falhas” nos modelos apresentados na seção anterior, pode-se notar uma grande contribuição para representar matematicamente o DARP. É interessante notar a complexidade dos modelos apresentados, o que conseqüentemente demonstra a complexidade do problema em si. Enfim, esses modelos deixam a desejar no que diz respeito a problemas reais, porém são excelentes pontos de partida para consolidar tal fato.

Neste trabalho, o DARP é tratado de forma estática (todas as requisições são conhecidas a priori), com múltiplos veículos, com frota heterogênea (cada veículo possui uma capacidade distinta) e garagens múltiplas (cada veículo começa e termina sua rota em garagens específicas). Esta é uma abordagem geral que abrange diversas outras encontradas na literatura (CORDEAU, 2006; CORDEAU; LAPORTE, 2003a; CORDEAU; LAPORTE, 2003b; JORGENSEN et al., 2007) e é bem próxima às situações reais do problema.

É proposto um modelo multi-objetivo (ver Seção 2.1) que busca minimizar simultaneamente os custos operacionais e a “insatisfação” dos clientes, ou seja, minimizar os requisitos “não-essenciais” do problema. É interessante destacar que, ao contrário dos modelos descritos no capítulo anterior, nesse modelo os requisitos essenciais também são colocados na função objetivo (3.41 e 3.42), o que representa uma “relaxação” para o problema. Essa idéia é interessante, pois torna mais amena a obtenção de uma solução para o problema. Entretanto, essa abordagem pode resultar em soluções inválidas.

Os requisitos não-essenciais (3.40) estão relacionados à distância total percorrida pelos veículos, ao número de veículos utilizados na solução do problema, à duração das rotas, ao tempo de viagem dos clientes e ao tempo de espera nos locais de embarque e desembarque. Já os requisitos essenciais são os mesmos descritos na Seção 3.2. O modelo é apresentado a seguir:

Minimizar:

$$\omega_0 \sum_{k \in K} \sum_{i \in N} \sum_{j \in N; j \neq i} (d_{i,j} x_{i,j}^k) + \omega_1 \sum_{k \in K} \sum_{j \in P} x_{g_k^-, j}^k + \omega_2 \sum_{k \in K} (B_{g_k^+} - D_{g_k^-}) + \omega_3 \sum_{i \in P} R_i + \omega_4 \sum_{i \in \{PUU\}} W_i + \quad (3.40)$$

$$\rho_0 \sum_{k \in K} \max\{0, (B_{g_k^+} - D_{g_k^-}) - T_k\} + \rho_1 \sum_{i \in P} \max\{0, R_i - R_i\} + \rho_2 \sum_{i \in \{PUU\}} \max\{0, W_i - W_i\} + \quad (3.41)$$

$$\rho_3 \sum_{k \in K} \max\{0, \left(Q_i \sum_{i \in \{PUU\}} \sum_{j \in \{PUU\}; j \neq i; j \neq n-i} x_{i,j}^k \right) - Q_k\} + \rho_4 \sum_{i \in N} (\max\{0, e_i - B_i\} + \max\{0, B_i - l_i\}) \quad (3.42)$$

Sujeito a:

$$\sum_{j \in \{PU\{g_k^+\}\}} x_{g_k^-, j}^k = 1 \quad \forall k \in K \quad (3.43)$$

$$\sum_{i \in \{UU\{g_k^-\}\}} x_{i, g_k^+}^k = 1 \quad \forall k \in K \quad (3.44)$$

$$\sum_{k \in K} \sum_{j \in \{PUU\}; j \neq i} x_{i,j}^k = 1 \quad \forall i \in P \quad (3.45)$$

$$\sum_{j \in \{PUU\}; j \neq i} x_{i,j}^k - \sum_{j \in \{PUUU\{g_k^+\}\}; j \neq i; j \neq n+i} x_{n+i,j}^k = 0 \quad \forall k \in K; i \in P \quad (3.46)$$

$$\sum_{j \in \{PUUU\{g_k^-\}\}; j \neq i; j \neq n+i} x_{j,i}^k - \sum_{j \in \{PUU\}; j \neq i} x_{i,j}^k = 0 \quad \forall k \in K; i \in P \quad (3.47)$$

$$\sum_{j \in \{PUU\}; j \neq i} x_{j,i}^k - \sum_{j \in \{PUUU\{g_k^+\}\}; j \neq i; j \neq n-i} x_{i,j}^k = 0 \quad \forall k \in K; i \in U \quad (3.48)$$

$$B_j = (B_i + s_i + t_{i,j} + W_j) \sum_{k \in K} x_{i,j}^k \quad \forall i, j \in N; i \neq j \quad (3.49)$$

$$Q_j = (Q_i + q_j) \sum_{k \in K} x_{i,j}^k \quad \forall i, j \in N; i \neq j \quad (3.50)$$

$$A_i = B_i - W_i \quad \forall i \in \{PUUU \cup G^+\} \quad (3.51)$$

$$D_i = B_i + s_i \quad \forall i \in \{PUUU \cup G^-\} \quad (3.52)$$

$$R_i = B_{n+i} - D_i \quad \forall i \in P \quad (3.53)$$

$$A_{g_k^-} = D_{g_k^+} = Q_{g_k^-} = Q_{g_k^+} = W_{g_k^-} = 0 \quad \forall k \in K \quad (3.54)$$

$$A_i, W_i, B_i, D_i, Q_i \quad \textit{irresritas} \quad \forall i \in N \quad (3.55)$$

$$R_i \quad \textit{irresrita} \quad \forall i \in P \quad (3.56)$$

$$x_{i,j}^k \in \{0, 1\} \quad \forall k \in K; \forall i, j \in N; i \neq j \quad (3.57)$$

A função objetivo é dividida em duas partes. A primeira (3.40) visa minimizar os requisitos não-essenciais do problema, enquanto a segunda (3.41 e 3.42) visa à minimização das violações nos requisitos essenciais. Além disso, os requisitos não-essenciais são “penalizados” por um vetor de números inteiros positivos (pesos) $\omega = [\omega_0, \omega_1, \omega_2, \omega_3, \omega_4]$, e os essenciais com um vetor semelhante $\rho = [\rho_0, \rho_1, \rho_2, \rho_3, \rho_4]$.

As restrições 3.43 e 3.44 garantem, respectivamente, que cada veículo partirá de sua garagem de origem e chegará a sua garagem de destino uma única vez, ou seja, cada rota será iniciada em sua garagem de origem e terminará em sua garagem de destino. A restrição 3.45 garante que cada cliente será atendido uma única vez, ou seja, por apenas um veículo. A restrição 3.46 garante que um local de embarque estará sempre na mesma rota que seu respectivo local de desembarque, enquanto que a contenção do fluxo, ou seja, tudo que entra é igual a tudo que sai, é garantida pelas restrições 3.47 e 3.48.

A restrição 3.49 determina o horário de início do serviço e o tempo de espera em cada local e também o veículo que o atenderá. Como mostrado em Cordeau (2006), essas restrições, em conjunto, garantem a eliminação de sub-rotas (*sub-tours*) no caso do DARP. A restrição 3.50 determina a carga dos veículos em cada local, enquanto as restrições 3.51, 3.52 e 3.53 garantem, respectivamente, um cálculo correto dos horários de chegada e partida nos locais e dos tempos de viagem dos clientes. A restrição 3.54 “inicializa” algumas variáveis referentes às garagens. Por fim, a restrição 3.57 apenas garante que as variáveis de decisão $x_{i,j}^k$ sejam binárias, enquanto as restrições 3.55 e 3.56 garantem que as demais variáveis sejam irrestritas.

Como pode ser observado no modelo proposto, devido às restrições 3.49 e 3.50, e a parte da função objetivo que representa os requisitos essenciais (3.41 e 3.42), esta formulação é não-linear. Entretanto, essas restrições podem ser linearizadas de forma análoga a utilizada por Cordeau (2006) (ver Subseção 3.2.1). Já os termos 3.41 e 3.42 podem ser substituídos por outras restrições, assim como nos modelos descritos

nas seções anteriores, porém passarão a representar o problema de uma forma não-relaxada. Para isso, basta remover os termos 3.41 e 3.42 da função objetivo, e inserir as seguintes restrições no modelo:

$$B_{g_k^+} - D_{g_k^-} \leq T_k \quad \forall k \in K \quad (3.58)$$

$$R_i \leq R_i \quad \forall i \in P \quad (3.59)$$

$$W_i \leq W_i \quad \forall i \in \{P \cup U\} \quad (3.60)$$

$$Q_i \leq Q_k \quad \forall i \in P; k \in K \quad (3.61)$$

$$e_i \leq B_i \leq l_i \quad \forall i \in N \quad (3.62)$$

Como sugerido em Cordeau (2006), nesse modelo é feita uma redução no número de arcos (caminhos) existentes entre os locais (como pode ser visto na formulação matemática). Tal fato se deve às características do DARP, que por exemplo, não permite um arco ligando um local a si próprio, não permite que um desembarque seja realizado antes de seu respectivo embarque, etc. Além disso, ao contrário da formulação proposta por Bergvinsdottir (2004), que utiliza as variáveis referentes aos locais de forma exclusiva para cada veículo $(A_i^k, B_i^k, D_i^k, W_i^k, Q_i^k, R_i^k)$, nessa formulação são utilizadas variáveis únicas $(A_i, B_i, D_i, W_i, Q_i, R_i)$, o que reduz significativamente o número de variáveis de decisão do modelo.

Este modelo “simplifica” a representação do problema, pois utiliza a redução no número de variáveis e restrições e também a eliminação de arcos do problema proposta por Cordeau (2006). Além disso, essa abordagem é mais condizente com a realidade, pois trata o problema com frota heterogênea, garagens múltiplas e com uma função multi-objetivo, o que possibilita a busca por soluções equilibradas entre os custos operacionais e a inconveniência dos clientes.

Por fim, vale ressaltar que esse modelo ainda pode ser aprimorado, como por exemplo, pela redução do número de variáveis e restrições, da linearização do modelo relaxado, e outras técnicas, possibilitando provavelmente a sua resolução por meio de um *software* comercial.

3.4 *Simulated Annealing* aplicado ao DARP

Um método heurístico baseado na aplicação do *Simulated Annealing* - SA (ver Seção 2.5), juntamente com outras heurísticas, é proposto para solucionar o DARP representado pelo modelo descrito na seção anterior. Para a utilização do SA, deve-se definir a priori um método para geração de uma solução inicial S , um método para geração das soluções vizinhas S' (estrutura de vizinhança), e uma função objetivo $f(S)$ a ser otimizada.

3.4.1 Geração da solução inicial

A solução inicial é gerada por uma “heurística de distribuição”, que é responsável pela roteirização dos veículos, ou seja, pela formação dos agrupamentos dos locais nas rotas e da sequência de atendimento destes. Essa heurística é apresentada na Figura 3.2.

```
1. CRIAR (m rotas vazias, e atribuí-las aos m veículos);
2. CRIAR (uma lista L com todas as requisições de transporte dos clientes);
3. PARA (cada rota k, k = 1,2,...,m) FAÇA
4.   SELECIONAR ( [n/m] requisições de L);
5.   PARA (cada requisição selecionada) FAÇA
6.     Pos1 ← posição qualquer da rota k;
7.     Pos2 ← posição qualquer da rota k, porém posterior a Pos1;
8.     INSERIR (o ponto de embarque em Pos1);
9.     INSERIR (o ponto de desembarque em Pos2);
10.  FIM-PARA;
11.  INSERIR (a garagem de origem do veículo k no início da rota k);
12.  INSERIR (a garagem de destino do veículo k no fim da rota k);
13.  REMOVER (as requisições selecionadas de L);
14. FIM-PARA;
```

Figura 3.2 - Heurística de distribuição para o DARP.

Nessa heurística, são criadas inicialmente m rotas vazias, sendo atribuído, a cada uma delas, um veículo específico. Posteriormente, todas as requisições de transporte dos clientes (pontos de embarque e seus respectivos pontos de desembarque) são distribuídas aleatoriamente, porém de forma uniforme, a essas rotas, ou seja, as n requisições são divididas igualmente entre os m veículos. Em alguns casos a divisão de n por m não será inteira, e nesses casos a última rota deverá atender a um número superior de requisições. A seleção das requisições e das rotas que as atenderão também é aleatória.

As posições de inserção dos pontos (locais) na rota também são selecionadas de forma aleatória, mas sempre atendendo à restrição de precedência oriunda da requisição de

transporte do cliente em questão, ou seja, um local de embarque sempre deverá ser anterior ao seu respectivo local de desembarque. Por fim, devido ao fato de toda rota começar e terminar em garagens específicas (referentes ao veículo que irá executá-la), pode-se considerar que os locais referentes a estas garagens sempre serão alocados como primeiro e último pontos em cada uma das rotas criadas.

Como descrito anteriormente, essa heurística trata apenas a roteirização dos veículos, entretanto ainda deve-se fazer a programação destes, ou seja, determinar os horários de chegada nos locais, os horários de partida, etc. Para isso, é utilizada uma outra heurística, denominada “heurística de programação” (Figura 3.3), que é adaptada da apresentada em Cordeau e Laporte (2003a) e realiza a programação de uma forma a tentar reduzir as violações nas janelas de tempo, a duração das rotas, e os tempos de viagem dos clientes. Nessa heurística, é utilizado um conceito de “atraso”, proposto inicialmente por Savelsbergh (1992), que consiste basicamente em atrasar, na medida do possível, o horário de partida da garagem de origem e o início do serviço nos locais de embarque.

1. $B_0 \leftarrow e_0; \quad D_0 \leftarrow B_0;$
2. CALCULAR (A_i, B_i, W_i, D_i, Q_i para cada ponto $v_i \in V_k$ e $v_i \neq v_0$);
3. CALCULAR (F_0);
4. $B_0 \leftarrow e_0 + \min \left\{ F_0, \sum_{0 < p < z} W_p \right\}; \quad D_0 \leftarrow B_0;$
5. ATUALIZAR (A_i, B_i, W_i, D_i para cada ponto $v_i \in V_k$ e $v_i \neq v_0$);
6. CALCULAR (R_i para cada ponto $v_i \in V_k$ e $v_i \in P$);
7. PARA (cada ponto $v_i \in V_k$ e $v_i \in P$) FAÇA
8. CALCULAR (F_i);
9. $B_i \leftarrow B_i + \min \left\{ F_i, \sum_{i < p < z} W_p \right\};$
10. $D_i \leftarrow B_i + s_i; \quad W_i \leftarrow B_i - A_i;$
11. ATUALIZAR (A_j, B_j, W_j, D_j para cada ponto $v_j \in V_k$ e v_j posterior a v_i);
12. ATUALIZAR (R_j para cada ponto $v_j \in V_k$, $v_j \in P$ e v_{n+j} posterior a v_i);
13. FIM-PARA;

Figura 3.3 - Heurística de programação para o DARP.

Inicialmente, o horário de partida da garagem de origem é marcado para o horário de início da respectiva janela de tempo. A partir de então, os demais cálculos (horários de chegada, início do serviço, tempo de espera, horário de partida e carga do veículo) são realizados para todos os pontos seguintes na rota. Posteriormente é calculado o atraso para a partida da garagem de origem, e então o horário de partida é ajustado

de forma a não aumentar as violações nas janelas de tempo. É feita então uma atualização nos horários para todos os pontos posteriores à garagem de origem, e também é calculado o tempo de viagem dos clientes. Por fim, para cada ponto de embarque pertencente à rota em questão, é calculado o seu respectivo atraso (sendo o horário de início do serviço ajustado de forma a reduzir a duração da rota e o tempo de viagem dos clientes, e não aumentar as violações nas janelas de tempo), e os horários de todos os pontos posteriores são atualizados, assim como o tempo de viagem dos clientes cujo desembarque é posterior ao ponto de embarque em questão.

Considerando então uma rota $V_k = \{v_0, v_i, v_j, v_{n+j}, v_{n+i}, \dots, v_z\}$, executada pelo veículo k ($\forall k \in K$), tem-se que v_0 e v_z representam, respectivamente, as garagens de origem e de destino de k ($v_0 \in G^-; v_z \in G^+$), e os demais pontos representam os pontos de embarque e desembarque, sendo que o ponto v_{n+i} representa o ponto de desembarque referente ao ponto de embarque v_i , ou seja $v_i \in P$ e $v_{n+i} \in U$. Para uma rota V_k qualquer, o atraso é calculado da maneira apresentada nas Equações 3.63, 3.64 e 3.65, e sua programação é ilustrada na Figura 3.3.

$$F_i = \min_{i \leq j \leq z} \left\{ \sum_{i < p \leq j} W_p + \left(\min\{l_j - B_j, \tilde{R}_i - R_j\} \right)^+ \right\} \quad \text{se } i \in P \quad (3.63)$$

$$F_i = \min_{i \leq j \leq z} \left\{ \sum_{i < p \leq j} W_p + (l_j - B_j) \right\} \quad \text{se } i \in G^- \quad (3.64)$$

$$F_i = 0 \quad \text{se } i \in \{U \cup G^+\} \quad (3.65)$$

3.4.2 Estrutura de vizinhança

Como estrutura de vizinhança foram utilizados três diferentes movimentos de troca: *re-ordenar rota*, *re-alocar ponto* e *trocar pontos*. Esses movimentos são baseados em outros encontrados frequentemente nos trabalhos referentes ao DARP (BERGVINS-DOTTIR, 2004; CORDEAU; LAPORTE, 2003a; SAVELSBERGH, 1992).

É interessante destacar que nesses movimentos não são consideradas as garagens, pois essas são “fixas” em todas as rotas, e conseqüentemente suas posições não poderão ser alteradas.

O movimento *re-ordenar rota* consiste basicamente em selecionar uma rota qualquer pertencente à solução, selecionar um ponto qualquer nessa rota, selecionar uma nova

posição para esse ponto e trocar sua posição para a nova. Esse movimento é ilustrado na Figura 3.4.

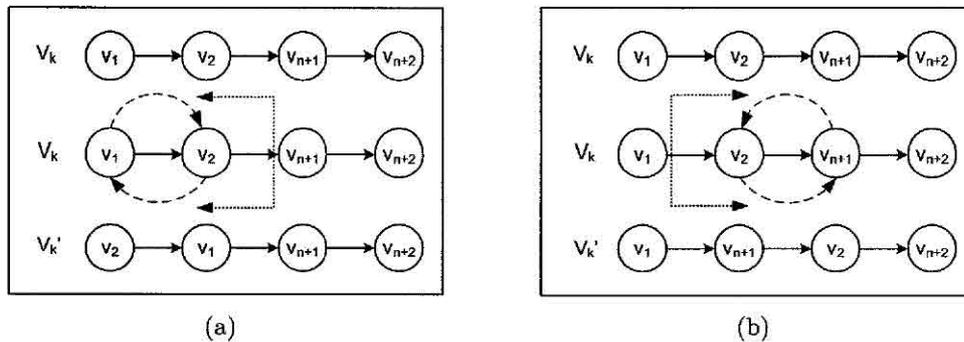


Figura 3.4 - Movimento re-ordenar rota.

Como é selecionado apenas um ponto na rota, este pode ser de embarque ou desembarque. No primeiro caso (Figura 3.4a), a nova posição deverá, obrigatoriamente, ser anterior ao seu respectivo ponto de desembarque. Já no segundo caso (Figura 3.4b), a nova posição deverá ser posterior ao respectivo ponto de embarque. Esses “limites” são apresentados pelas linhas pontilhadas na Figura 3.4.

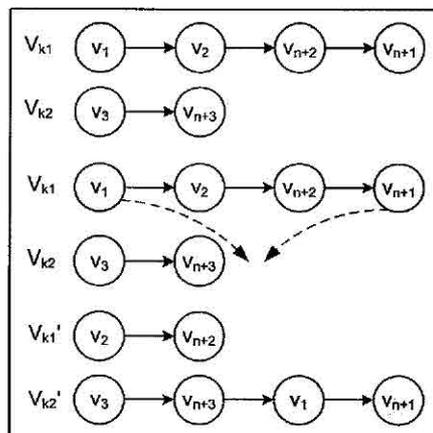


Figura 3.5 - Movimento re-alocar pontos.

O movimento *re-alocar pontos* consiste basicamente em selecionar duas rotas quaisquer pertencentes à solução, selecionar uma requisição qualquer em apenas uma das duas rotas, extraí-la (seus pontos de embarque e desembarque) de sua rota de origem e adicioná-la na outra rota, em posições quaisquer. O ponto de embarque e seu res-

pectivo ponto de desembarque são extraídos simultaneamente, porém sua inserção na “outra” rota pode ser feita de forma separada, ou seja, esses pontos são alocados individualmente em qualquer posição da rota, porém sempre mantendo a condição de que o ponto de embarque fique antes do ponto de desembarque (restrição de precedência). Esse movimento é ilustrado na Figura 3.5.

Já o movimento *trocar pontos* consiste basicamente em selecionar duas rotas quaisquer pertencentes à solução, selecionar uma requisição de transporte (ponto de embarque e seu respectivo ponto de desembarque) qualquer em cada uma das duas rotas, e trocá-las. Esse movimento é ilustrado na Figura 3.6. Nesse caso, como são trocadas requisições de transporte, ou seja, seus pontos de embarque e desembarque são trocados simultaneamente (trocados em pares), pode-se garantir que o ponto de embarque será sempre anterior ao seu respectivo ponto de desembarque.

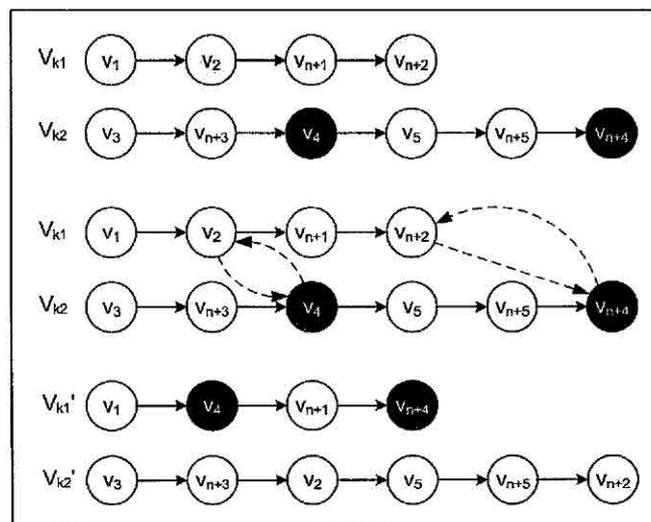


Figura 3.6 - Movimento trocar pontos.

O SA é implementado de forma análoga ao algoritmo apresentado na Seção 2.5, porém cada solução vizinha deverá ser gerada por apenas um desses movimentos, sendo a sua escolha feita de forma aleatória, porém uniformemente distribuída, possibilitando assim uma boa diversidade entre as soluções intermediárias geradas, e consequentemente uma boa exploração do espaço de soluções.

A função objetivo $f(S)$ utilizada para avaliar as soluções é aquela descrita pelas expressões 3.40, 3.41 e 3.42 do modelo descrito na Seção 3.3, e as restrições apresen-

tadas nesse mesmo modelo (3.43 a 3.57) são atendidas implicitamente nas heurísticas de distribuição e programação e nos movimentos de troca aqui descritos. Um pseudo-código do SA implementado é apresentado na Figura 3.7.

```

1. DADO ( $\alpha$ ,  $S_{\max}$ ,  $T_0$  e  $T_c$ ) FAÇA
2. GERAR (uma solução  $S$  através da heurística de distribuição);
3. APLICAR (a heurística de programação em todas as rotas de  $S$ );
4.  $S^* \leftarrow S$ ;           {Melhor solução obtida até então}
5.  $IterT \leftarrow 0$ ;       {Número de iterações na temperatura  $T$ }
6.  $T \leftarrow T_0$ ;        {Temperatura corrente}
7. ENQUANTO ( $T > T_c$ ) FAÇA
8.   ENQUANTO ( $IterT < S_{\max}$ ) FAÇA
9.      $IterT \leftarrow IterT + 1$ ;
10.    GERAR (um vizinho qualquer  $S'$  através de um dos mov. de troca);
11.    APLICAR (a heurística de programação em todas as rotas de  $S'$ );
12.     $\Delta \leftarrow f(S') - f(S)$ ;
13.    SE ( $\Delta < 0$ )  $S \leftarrow S'$ ;
14.    SE ( $f(S') < f(S^*)$ )  $S^* \leftarrow S'$ ; FIM-SE;
15.    SENÃO
16.      TOMAR ( $x \in [0,1]$ );
17.      SE ( $x < e^{-\Delta/T}$ )  $S \leftarrow S'$ ; FIM-SE;
18.    FIM-SE;
19.    FIM-ENQUANTO;
20.     $T \leftarrow \alpha * T$ ;  $IterT \leftarrow 0$ ;
21. FIM-ENQUANTO;
22.  $S \leftarrow S^*$ ;
23. RETORNAR ( $S$ );

```

Figura 3.7 - Algoritmo *Simulated Annealing* implementado para o DARP.

3.5 Experimentos computacionais

Vários experimentos foram realizados utilizando um conjunto de instâncias disponíveis na *OR-Library* (BEASLEY, 1990). Essas instâncias são referências nos trabalhos mais recentes e que têm considerável importância para resolução do DARP (CORDEAU; LAPORTE, 2003a; JORGENSEN et al., 2007). Estão disponíveis 20 instâncias, combinadas entre 24 a 144 requisições de transporte (48 a 288 pontos) e 3 a 13 veículos, sendo as primeiras instâncias (R1a - R10a) formadas por janelas de tempo “largas” e o restante (R1b - R10b) formado por janelas de tempo mais “estreitas”.

Essas instâncias representam problemas com garagem única e frota homogênea, e não adotam o conceito de tempo máximo de espera. Entretanto, o modelo aqui proposto (ver Seção 3.3) se adapta facilmente a elas. Para isso, basta tratar essa única garagem como garagem de origem e destino para todos os veículos, e um valor

muito grande como tempo máximo de espera (que irá eliminar a possibilidade de ocorrer qualquer violação).

3.5.1 CPLEX

Inicialmente, visando verificar a complexidade dos modelos já existentes (ver Subseções 3.2.1 e 3.2.2) e do modelo proposto (ver Seção 3.3), o *software* CPLEX (ILOG, 2006), que é uma referência para resolver problemas de otimização, foi utilizado para tentar solucionar a instância “mais simples” desse conjunto (R1a).

O modelo proposto por Cordeau (2006) foi implementado da forma como apresentada na Subseção 3.2.1, porém com as eliminações de arcos e as reduções no número de variáveis e restrições descritas na mesma seção. O custo representado na função objetivo do modelo em questão (3.1) foi dado pela da distância total percorrida pelos veículos, assim como no trabalho original de Cordeau (2006).

Já o modelo proposto por Bergvinsdottir (2004) foi implementado com as correções descritas na Subseção 3.2.2 (correções às falhas matemáticas do modelo), e foi considerada a mesma função objetivo utilizada no modelo proposto por Cordeau (2006), ou seja, apenas visando a minimização da distância total percorrida pelos veículos. Os demais termos da função objetivo (3.21 e 3.22) foram desconsiderados pela penalização nula, ou seja, com pesos iguais a zero.

Por fim, o modelo proposto também foi implementado de forma análoga as demais descritas acima, ou seja, com as simplificações e a função objetivo considerando apenas a distância, e tratando os requisitos essenciais do problema como restrições.

O resultado obtido confirmou a expectativa da complexidade do problema, pois o CPLEX não conseguiu resolver nenhum dos modelos, ou seja, não conseguiu obter nenhuma solução viável para a “menor” instância do problema.

3.5.2 *Simulated Annealing*

Os parâmetros utilizados pelo SA em todos os experimentos foram $T_0 = 20000$, $\alpha = 0.975$, $T_c = 0.01$ e $SAMax = 1000$. Já a escolha dos valores das penalizações (vetores ω e ρ) aplicadas na função objetivo (3.40, 3.41 e 3.42) foi baseada em uma análise apresentada por Bergvinsdottir (2004). Porém, neste trabalho a penalização para as violações nos requisitos essenciais foi mais “pesada”, visando evitar a obtenção de

soluções inválidas para o problema. Os pesos utilizados foram: $\omega = [8,0,1,3,1]$ e $\rho = [1500,1500,1500,1500,1500]$.

$$\text{Desvio} = \left(\frac{f(S)\text{média} - \text{Melhor } f(S)}{\text{Melhor } f(S)} \right) * 100 \quad (3.66)$$

Para validação do modelo proposto e da aplicação do SA ao DARP, foram realizados inicialmente cinco testes para cada instância. A Tabela 3.1 apresenta um resumo dos resultados obtidos ($f(S)$ é o valor da função objetivo obtida). Nessa tabela, a coluna *Melhor $f(S)$* indica o valor da função objetivo da melhor solução encontrada nos cinco testes (para cada instância). A coluna *$f(S)$ média* apresenta a média aritmética das cinco soluções encontradas, e a coluna *Desvio* apresenta o “desvio” obtido entre esses testes (ver Equação 3.66).

Tabela 3.1 - Resumo dos experimentos realizados para o DARP.

Instância	Nº de veículos	Nº de requisições	$f(S)$ média	Melhor $f(S)$	Desvio (%)
R1a	3	24	3721,57	3677,91	1,19
R2a	5	48	7101,87	7017,34	1,20
R3a	7	72	11982,18	11873,76	0,91
R4a	9	96	13982,52	13725,92	1,87
R5a	11	120	16006,82	15736,66	1,72
R6a	13	144	20607,77	20465,39	0,70
R7a	4	36	5718,46	5610,05	1,93
R8a	6	72	11554,71	11343,19	1,86
R9a	8	108	17345,60	15632,09	10,96
R10a	10	144	23104,61	22430,00	3,01
R1b	3	24	3407,87	3379,74	0,83
R2b	5	48	5925,35	5889,56	0,61
R3b	7	72	11045,26	11006,12	0,36
R4b	9	96	12856,75	12807,87	0,38
R5b	11	120	14874,48	14544,13	2,27
R6b	13	144	18795,79	18518,82	1,50
R7b	4	36	5202,93	5136,37	1,30
R8b	6	72	10791,67	10703,17	0,83
R9b	8	108	15180,91	15013,71	1,11
R10b	10	144	20492,56	19969,15	2,62

Em todos os experimentos realizados o número de veículos utilizados foi igual ao disponível, todas as requisições de transportes foram atendidas, e o mais importante,

todas as soluções obtidas foram válidas. Os melhores resultados obtidos (Tabela 3.2) ainda são comparados aos obtidos por Cordeau e Laporte (2003a), com a metaheurística Busca Tabu (Tabela 3.3), e por Jorgensen et al. (2007), com um Algoritmo Genético (Tabela 3.4). Apenas os resultados obtidos para algumas instâncias são comparados a esses trabalhos, pois Jorgensen et al. (2007) não apresentam os resultados para as demais.

Tabela 3.2 - Melhores soluções obtidas pelo *Simulated Annealing* para o DARP.

Inst.	Distância percorrida	Duração das rotas (min)	Tempo de espera (min)		Tempo de viagem (min)		CPU (min)
			total	médio	total	médio	
R1a	252,79	831,30	98,51	2,05	241,93	10,08	1,00
R2a	437,45	1992,34	594,90	6,20	310,17	6,46	1,20
R3a	831,74	2404,67	132,93	0,92	894,08	12,42	1,46
R5a	1085,45	3920,25	434,81	1,81	899,35	7,49	1,79
R9a	1064,23	3258,66	34,42	0,16	1275,06	11,81	2,28
R10a	1392,09	4475,42	203,33	0,71	2204,85	15,31	2,72
R1b	251,85	738,42	6,57	0,14	206,66	8,61	0,92
R2b	436,69	1428,44	31,75	0,33	311,95	6,50	1,30
R5b	1010,09	3654,02	243,94	1,02	855,16	7,13	1,95
R6b	1289,31	4318,33	149,02	0,52	1245,66	8,65	1,94
R7b	375,67	1095,67	0	0	345,10	9,59	1,05
R9b	1041,09	3315,28	114,19	0,53	1085,18	10,05	2,26
R10b	1414,65	4332,69	38,04	0,13	1427,08	9,91	2,77
TOTAL	10883,10	35765,49	2082,41	14,52	11302,23	124,01	22,64

Comparando os resultados obtidos com os apresentados por Cordeau e Laporte (2003a) (Tabela 3.3), a abordagem proposta neste trabalho apresentou um aumento de 63,53% na distância total percorrida pelos veículos, porém no tempo de duração das rotas houve um aumento de apenas 0,64%. Já no tempo médio de espera dos veículos, esta abordagem foi capaz de obter uma redução de 66,17%, no tempo médio de viagem dos clientes a redução foi de 80,46%, e o mais interessante, 95,45% de redução no tempo de processamento.

Em relação aos resultados obtidos por Jorgensen et al. (2007) (Tabela 3.4), a abordagem proposta neste trabalho foi capaz de reduzir 11,71% no tempo de duração das rotas, 48,53% no tempo médio de espera dos veículos, 75,33% no tempo médio de viagem dos clientes, e 95,37% no tempo de processamento.

Tabela 3.3 - Melhores soluções obtidas pela Busca Tabu para o DARP.

Inst.	Distância percorrida	Duração das rotas (min)	Tempo de espera (min)		Tempo de viagem (min)		CPU (min)
			total	médio	total	médio	
R1a	190,02	881,16	211,15	4,40	1094,99	45,62	1,90
R2a	302,08	1985,94	723,87	7,54	1976,73	41,18	8,06
R3a	532,08	2579,35	607,27	4,22	3586,68	49,82	17,18
R5a	636,97	3869,95	832,98	3,47	6156,48	51,30	46,24
R9a	672,44	3155,49	323,05	1,50	5621,77	52,05	50,51
R10a	878,76	4480,10	721,33	2,50	7163,71	49,75	87,53
R1b	164,46	965,06	320,60	6,68	1041,50	43,40	1,93
R2b	296,06	1564,74	308,68	3,22	2393,18	49,86	8,29
R5b	589,74	3595,63	605,89	2,52	6104,72	50,87	54,33
R6b	743,60	4072,47	448,88	1,56	7347,39	51,02	73,70
R7b	248,21	1097,25	129,03	1,79	1761,99	48,94	4,23
R9b	601,96	3249,29	487,33	2,26	5581,02	51,68	51,28
R10b	798,63	4040,99	362,37	1,26	7072,29	49,11	92,41
TOTAL	6655,01	35537,42	6082,43	42,92	56902,45	634,60	497,59

Tabela 3.4 - Melhores soluções obtidas pelo Algoritmo Genético para o DARP.

Inst.	Distância percorrida	Duração das rotas (min)	Tempo de espera (min)		Tempo de viagem (min)		CPU (min)
			total	médio	total	médio	
R1a	-	1039	260	5,42	310	12,90	5,57
R2a	-	1994	514	5,36	1330	27,72	11,43
R3a	-	2781	301	2,09	2894	40,20	21,58
R5a	-	4274	527	2,20	4837	40,30	58,23
R9a	-	3526	32	0,15	6719	62,21	40,78
R10a	-	5025	246	0,86	8341	57,92	65,98
R1b	-	928	164	3,42	549	22,89	5,46
R2b	-	1710	162	1,69	1300	27,07	11,72
R5b	-	4336	568	2,37	4720	39,33	58,93
R6b	-	5227	513	1,78	6397	44,42	81,23
R7b	-	1316	128	1,78	784	21,76	8,29
R9b	-	3676	177	0,82	5358	49,61	44,66
R10b	-	4678	85	0,29	8119	56,38	66,41
TOTAL	-	40508	3678	28,21	51657	502,72	488,61

Como as distâncias percorridas pelos veículos não são apresentadas em Jorgensen et al. (2007), não é feita nenhuma comparação. Já as distâncias apresentadas em Cordeau e Laporte (2003a), e neste trabalho, são distâncias Euclidianas entre os pontos, e conseqüentemente não apresentam unidade de medida.

3.6 Considerações finais

Analisando os trabalhos já existentes sobre o DARP, nota-se que, além dos diferentes métodos para sua resolução, existem inúmeras variações nas formulações. Algumas dessas variações dizem respeito à definição de frota múltipla ou veículo único, de frota homogênea ou heterogênea, de múltiplas garagens ou garagem única, e principalmente de diversas formulações para a função objetivo que se deseja otimizar.

A formulação matemática proposta pode ser facilmente utilizada para solucionar diversos outros modelos já conhecidos para o DARP, como por exemplo, modelos com garagem única, com apenas um veículo, com frota homogênea, e o mais interessante, com vários objetivos distintos. Essa flexibilidade na escolha do objetivo final é dada pela utilização de uma função multi-objetivo, que utiliza pesos para determinar a “importância” de cada requisito na solução do problema.

É interessante destacar também que essa formulação é baseada em várias outras encontradas na literatura (BERGVINSDOTTIR, 2004; CORDEAU, 2006; CORDEAU; LAPORTE, 2003a). Além disso, o modelo representa o problema de forma “relaxada”, tratando os requisitos essenciais como parte da função a ser minimizada, e não como restrições do modelo. Essa é uma abordagem interessante que torna a resolução do problema menos árdua, porém permite que soluções inválidas sejam obtidas. Entretanto, soluções inválidas podem ser interessantes em alguns casos, pois como é sabido (CORDEAU, 2006), os modelos exatos são incapazes de resolver de forma satisfatória um problema de tamanho “real”, e em alguns casos uma solução com violações mínimas nos requisitos essenciais pode ser aceitável.

A escolha do *Simulated Annealing* se deve aos excelentes resultados obtidos para diversos problemas de otimização (BAUGH et al., 1998; BEASLEY, 1998; MAURI, 2003; MAURI; LORENA, 2006b; MAURI et al., 2008b), e também a sua simplicidade. Já a escolha das demais heurísticas e da estrutura de vizinhança foi baseada em outras aplicações das mesmas ao DARP.

O SA integrado com as demais heurísticas apresentadas na Seção 3.4 foi capaz de

obter, em todos os casos e com pouco tempo de processamento, soluções válidas para o problema. O SA se mostrou robusto, pois como pode ser observado na Tabela 3.1, os desvios apresentados foram satisfatoriamente baixos. Além disso, a estrutura de vizinhança utilizada (movimentos de troca) mostrou ser adequada e eficiente para exploração do espaço de soluções.

Os resultados obtidos mostram que o *Simulated Annealing* juntamente com o modelo proposto e as demais heurísticas descritas na Seção 3.4 foram capazes de gerar soluções de boa qualidade para todas as instâncias em tempos computacionais expressivamente baixos. Esses resultados ainda foram comparados com duas outras abordagens recentes encontradas na literatura, e em todos os casos a “qualidade do serviço” foi expressivamente maior, ou seja, a inconveniência dos clientes foi significativamente reduzida, o que na prática, em se tratando da perspectiva humana, reflete melhores soluções.

Por fim, vale destacar que a maioria das instâncias utilizadas nos demais trabalhos descritos neste capítulo são obtidas por intermédio de empresas que atuam no setor, que na maioria dos casos não as disponibilizam publicamente, o que impossibilita comparações diretas com outros métodos encontrados na literatura. Sendo assim, buscou-se conseguir junto à Secretaria de Transportes do município de São José dos Campos - SP um conjunto de dados reais referentes ao serviço de transporte de PPNEs que opera neste município, entretanto notou-se uma grande “resistência” dos responsáveis por este serviço em ceder tais dados para essa pesquisa.

Todos os experimentos apresentados neste capítulo foram realizados em um *laptop Toshiba A10 S127* com processador *Intel Celeron®* de 2.0 GHz e 256MB de memória RAM. Toda a implementação foi desenvolvida na linguagem C++ com chamadas à biblioteca do *software* CPLEX 10.0. As soluções obtidas por Cordeau e Laporte (2003a) foram executadas em um *PC Intel Pentium 4* de 2.0 GHz, e as obtidas por Jorgensen et al. (2007) em um *PC Celeron®* de 2.0 GHz.

4 PROBLEMA DE ALOCAÇÃO DE BERÇOS

O Problema de Alocação de Berços (PAB) consiste em atribuir os navios que chegam a um determinado porto para as “posições” de atracação disponíveis ao longo de um cais (berços). As principais decisões a serem tomadas neste processo envolvem a escolha de onde e quando os navios deverão atracar (CORDEAU et al., 2005).

Em relação ao local de atracação, existem restrições relativas à profundidade da água, à distância máxima em relação ao local mais favorável ao longo do cais, e também ao tamanho dos navios. Já em relação ao horário de atracação dos navios, as restrições são expressas como janelas de tempo para conclusão de seu atendimento (CORDEAU et al., 2005).

O tempo de atendimento de um navio depende de seu ponto de atracação (berço), e é uma função da distância do berço até a área de carga e descarga de contêineres no pátio do porto. Como mencionado em Cordeau et al. (2005), esta dependência afeta fortemente o desempenho das operações no porto. O cenário onde ocorre o problema de alocação de berços é ilustrado pela Figura 4.1.

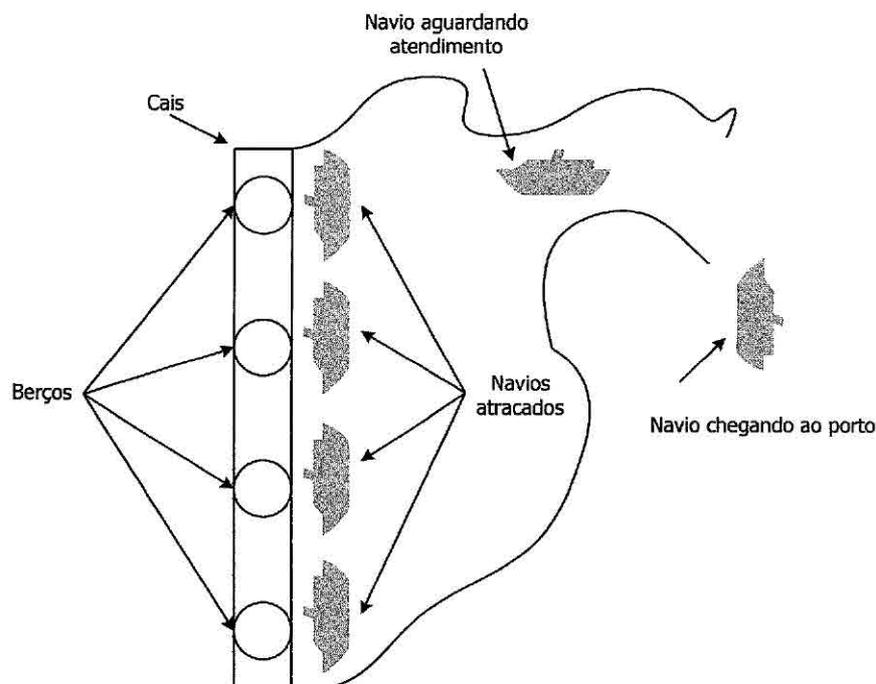


Figura 4.1 - Cenário típico para o PAB.

Os dados referentes ao tempo de serviço dependem de outra decisão, que é o número de guindastes no cais que estão disponíveis aos navios que chegam. Essa “decisão” é conhecida como Problema de Atribuição de Guindastes - PAG (LEE et al., 2008). Esta decisão afeta o tempo de serviço dos navios, e conseqüentemente tem um impacto no PAB. Em um sistema complexo, como um porto de transferência de cargas, o processo de tomada de decisão é frequentemente hierárquico, e o PAG é resolvido antes do PAB.

O objetivo então é minimizar os custos referentes ao porto e ao navio, o que é geralmente relacionado ao tempo de serviço dos navios. O objetivo do PAB normalmente é minimizar o tempo total de serviço de todos os navios. Desde que os navios não tenham a mesma importância, uma soma dos tempos de serviços dos navios, considerando uma penalização para indicar a sua devida importância, pode refletir melhor a prática de gerenciamento de alguns portos. Os pesos nesta soma podem representar valores estimados da carga ou do número de contêineres movimentados. Em algumas variantes do problema, como em Inai et al. (2008) por exemplo, também podem ser incluídas outras condições de penalidade na função objetivo.

O PAB pode ser modelado como um problema discreto se o cais for visto como um conjunto finito de berços. Neste caso, os berços podem ser descritos como segmentos de comprimento fixos, ou, se a dimensão de espaço for ignorada, como pontos. Já os modelos contínuos consideram que os navios podem atracar em qualquer lugar ao longo do cais. No caso discreto, o PAB pode ser modelado como o Problema de Programação de Máquinas Paralela Sem Conexão (PINEDO, 2001), onde um navio é tratado como um trabalho e um berço como uma máquina. No caso contínuo, o PAB é representado como um Problema de Corte de Estoque Bi-dimensional com algumas restrições adicionais. Em ambos os casos, o PAB é *NP-Hard* (GAREY; JOHNSON, 1979). Além dessas abordagens, o PAB pode ser tratado de forma estática ou dinâmica. Nessa primeira, considera-se que todos os navios já estão no porto para serem atendidos. Já na segunda, que reflete melhor as situações reais de um porto, considera-se que os navios deverão chegar ao porto ao longo do dia em horários distintos (STEENKEN et al., 2004).

4.1 Revisão da literatura

Os principais trabalhos referentes ao PAB são baseados em métodos heurísticos. Tal fato se deve principalmente ao tratamento de problemas reais, que aparecem

na maioria desses trabalhos. Assim como o DARP (ver Capítulo 3), o PAB é um problema ainda pouco explorado, e os principais trabalhos que o trata são recentes.

Imai et al. (1997) propõem uma formulação matemática e uma heurística para o PAB considerando que todos os navios estejam no porto antes da abertura dos berços (PAB estático). Essa abordagem não condiz com a realidade, pois os navios devem chegar em horários diferentes ao longo do dia (PAB dinâmico). Porém, esse trabalho é um dos pioneiros a respeito do PAB. Uma extensão deste trabalho, que trata o PAB dinâmico, é apresentada em Imai et al. (2001). Nesse trabalho, é utilizada uma heurística baseada na Relaxação Lagrangiana para resolver instâncias com 25 e 50 navios e 5, 7 e 10 berços.

Imai et al. (2003) utilizam a mesma heurística apresentada em Imai et al. (2001), porém para uma versão do PAB que considera “prioridades” para o atendimento dos navios. Nessa versão, o problema é modelado como um *Problema Quadrático de Atribuição*, e um Algoritmo Genético é proposto para resolvê-lo. Os autores utilizam instâncias com 25, 50, 75 e 150 navios e 5 berços. Imai et al. (2008) apresentam uma nova formulação para o PAB, dessa vez considerando a possibilidade de utilização de “berços” extras para o atendimento dos navios. Mais uma vez, eles utilizam um AG para resolver o problema. É interessante destacar que vários portos (reais), de várias partes do mundo, são considerados para geração das instâncias utilizadas nesses trabalhos.

Kim e Moon (2003) apresentam uma formulação de programação linear inteira mista para o caso contínuo do PAB. Os autores utilizam o *solver LINDO* e propõem uma heurística baseada no *Simulated Annealing* para resolver instâncias reais (de um porto Coreano) com até 40 navios. O *LINDO* apresenta resultados apenas para as instâncias com até 7 navios. Já o SA apresenta boas soluções para todas as instâncias.

Cordeau et al. (2005) apresentam uma heurística baseada na Busca Tabu para resolver duas formulações distintas para os casos discreto e contínuo do PAB. Também são propostas duas heurísticas, uma para resolver o caso contínuo do problema e uma para o caso discreto. Os autores mostram que apenas instâncias de pequeno porte podem ser resolvidas de forma exata, e nesses casos, a Busca Tabu proposta sempre encontra as mesmas soluções (ótimas). Já para instâncias de grande porte, a Busca Tabu sempre apresenta resultados melhores do que uma versão “truncada” de um *branch-and-bound* aplicado a uma formulação linear. As formulações propostas

e a Busca Tabu ainda são capazes de tratar várias características encontradas em problemas reais, como por exemplo, janelas de tempo e berços “favoritos” de atracação. Os métodos propostos são aplicados a um grande conjunto de instâncias, sendo a maioria gerada com base em informações reais do porto de Gioia Tauro - Itália, e 30 delas geradas aleatoriamente, com 60 navios e 13 berços.

Hansen et al. (2008) apresentam uma heurística de busca em vizinhança variável (VNS), uma heurística VNS com inicializações múltiplas (*multi-start*), um Algoritmo Genético, e um Algoritmo Memético para resolver o caso discreto do PAB. Esses métodos são aplicados a vários conjuntos de instâncias, variando de 50 navios e 5 berços até 200 navios e 20 berços. Os resultados obtidos pelo VNS superam os demais. Para várias dessas instâncias o CPLEX apresentou as soluções ótimas.

Vários casos que tratam o PAB em sua versão contínua são apresentados em Nishimura et al. (2001), Kim e Moon (2003), Imai et al. (2005), Cordeau et al. (2005), etc. Outra abordagem baseada na Relaxação Lagrangiana é apresentada em Monaco e Sammarra (2007). Um modelo de simulação das atividades portuárias é apresentado em Dragovic et al. (2005). Outros problemas relativos ao gerenciamento portuário são apresentados em Gunther e Kim (2006), Cordeau et al. (2007) e Lee et al. (2008). Vis e Koster (2003) apresentam uma discussão a respeito desses problemas. Por fim, Steenken et al. (2004) apresentam uma classificação dos principais processos e operações presentes em um porto.

4.2 Formulação matemática

Assim como apresentado em Imai et al. (2001), Imai et al. (2003) e Cordeau et al. (2005), neste trabalho o problema é tratado em sua forma discreta e dinâmica, considerando como objetivo principal a minimização do tempo total gasto pelos navios dentro do porto, o que segundo Hansen et al. (2008) é uma função objetivo apropriada para o PAB. Dessa forma, o cais é dividido em um conjunto finito de berços, e a dimensão espacial é ignorada. Logo, deve-se observar, para cada navio, os tempos descritos na Figura 4.2.

Como observado por Legato et al. (2001), o PAB pode ser modelado como um *Problema de Roteamento de Veículos com Garagens Múltiplas e Janelas de Tempo*. Neste trabalho, o PAB é representado inicialmente de acordo com o modelo matemático proposto por Cordeau et al. (2005).

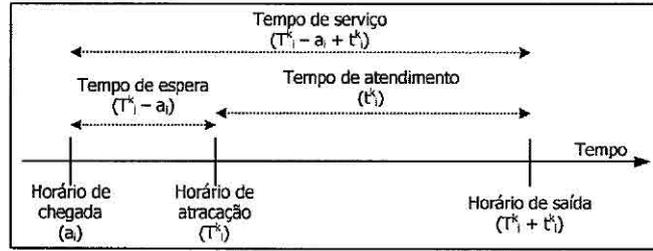


Figura 4.2 - Variáveis referentes ao tempo.

Nesse modelo, os navios são tratados como clientes e os berços como garagens (cada uma com seu veículo específico). Existem então m veículos “fictícios” (um para cada garagem), sendo que cada um inicia e termina sua “rota” na sua própria garagem. Os navios são modelados como vértices em um multi-grafo, onde cada garagem (berço) ainda é dividida em um vértice de origem e um de destino. Nos vértices de origem e destino, as janelas de tempo correspondem ao período de funcionamento dos berços.

O modelo então é dado por um multi-grafo $G^k = (V^k, A^k)$, $\forall k \in M$, onde $V^k = N \cup \{o(k), d(k)\}$ e $A^k \subseteq V^k \times V^k$. As variáveis e constantes usadas para representar o problema são:

- N : conjunto de navios, $n = |N|$;
- M : conjunto de berços, $m = |M|$;
- t_i^k : duração do atendimento do navio i no berço k ;
- a_i : horário de chegada do navio i ;
- s^k : horário de abertura do berço k ;
- e^k : horário de fechamento do berço k ;
- b_i : horário de término da janela de tempo para o navio i ;
- v_i : valor (custo) do tempo de serviço do navio i ;
- $x_{ij}^k \in \{0, 1\}, \forall k \in M, \forall (i, j) \in A^k, x_{ij}^k = 1$ se o navio j é atendido pelo berço k após o navio i ;
- $T_i^k \forall k \in M, i \in N$ é o horário em que o navio i atracou no berço k ;
- $T_{o(k)}^k \forall k \in M$ é o horário em que o primeiro navio atracou no berço k ;

- $T_{d(k)}^k \forall k \in M$ é o horário em que o último navio saiu do berço k ;
- $M_{ij}^k = \max\{b_i + t_i^k - a_j, 0\}, \forall k \in M, \forall (i, j) \in N$.

O modelo do PAB proposto por Cordeau et al. (2005) é escrito da seguinte forma:

Minimizar:

$$\sum_{i \in N} \sum_{k \in M} v_i \left[T_i^k - a_i + t_i^k \sum_{j \in NU\{d(k)\}} x_{ij}^k \right] \quad (4.1)$$

Sujeito a:

$$\sum_{k \in M} \sum_{j \in NU\{d(k)\}} x_{ij}^k = 1 \quad \forall i \in N \quad (4.2)$$

$$\sum_{j \in NU\{d(k)\}} x_{o(k)j}^k = 1 \quad \forall k \in M \quad (4.3)$$

$$\sum_{i \in NU\{o(k)\}} x_{id(k)}^k = 1 \quad \forall k \in M \quad (4.4)$$

$$\sum_{j \in NU\{d(k)\}} x_{ij}^k - \sum_{j \in NU\{o(k)\}} x_{ji}^k = 0 \quad \forall k \in M, \forall i \in N \quad (4.5)$$

$$T_i^k + t_i^k - T_j^k \leq (1 - x_{ij}^k) M_{ij}^k \quad \forall k \in M, \forall (i, j) \in A^k \quad (4.6)$$

$$T_i^k \geq a_i \quad \forall k \in M, \forall i \in N \quad (4.7)$$

$$T_i^k + t_i^k \sum_{j \in NU\{d(k)\}} x_{ij}^k \leq b_i \quad \forall k \in M, \forall i \in N \quad (4.8)$$

$$T_{o(k)}^k \geq s^k \quad \forall k \in M \quad (4.9)$$

$$T_{d(k)}^k \leq e^k \quad \forall k \in M \quad (4.10)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall k \in M, \forall (i, j) \in A^k \quad (4.11)$$

A função objetivo minimiza o tempo decorrido desde o momento em que os navios chegam, atracam e são atendidos, considerando um custo de serviço para esse tempo. A restrição 4.2 garante que cada navio é atendido por apenas um berço. As restrições 4.3 e 4.4 garantem, respectivamente, que um navio será o primeiro a ser atendido em cada berço, e outro será o último. A restrição 4.5 garante a conservação do fluxo (atendimento) para os demais navios. A restrição 4.6 faz o cálculo do horário de

atracação dos navios. Nessa restrição são considerados apenas os arcos A^k válidos para cada berço k , ou seja, alguns navios não podem ser atendidos em determinados berços, pois por exemplo, o tipo de equipamento disponível no berço pode não ser apropriado para o atendimento de determinados tipos de carga. A possibilidade de atendimento ou não dos navios pelos berços é determinada nas instâncias utilizadas. As restrições 4.7 e 4.8 garantem, respectivamente, que o horário de atracação seja após a chegada do navio, e que o horário do término do atendimento do navio seja anterior ao horário limite do navio (janela de tempo). As restrições 4.9 e 4.10 garantem a não violação das janelas de tempo nos berços. A restrição 4.11 garante que as variáveis de decisão sejam binárias.

4.3 Modelo proposto

Para utilização dos métodos apresentados nas próximas seções, é proposto um modelo matemático baseado na formulação descrita na seção anterior. Esse modelo é uma relaxação do apresentado por Cordeau et al. (2005).

Nesse modelo, as restrições 4.7 e 4.8 foram relaxadas, sendo transferidas para a função objetivo (4.13). De forma análoga, as restrições 4.9 e 4.10 também foram transferidas para a função objetivo (4.14). As demais restrições foram mantidas, porém, na função objetivo foram adicionados fatores de penalização ($\omega = [\omega_0, \omega_1, \omega_2]$) para cada termo. O modelo proposto é apresentado a seguir.

Minimizar:

$$\omega_0 \sum_{i \in N} \sum_{k \in M} v_i \left(T_i^k - a_i + t_i^k \sum_{j \in NU\{d(k)\}} x_{ij}^k \right) + \quad (4.12)$$

$$w_1 \sum_{i \in N} \sum_{k \in M} \left(\max(0, a_i - T_i^k) + \left(\max \left(0, T_i^k + t_i^k \sum_{j \in NU\{d(k)\}} x_{ij}^k - b_i \right) \right) \right) \quad (4.13)$$

$$\omega_2 \sum_{k \in M} \left(\max(0, s^k - T_{o(k)}^k) + \max(0, T_{d(k)}^k + e^k) \right) \quad (4.14)$$

Sujeito a:

$$\sum_{k \in M} \sum_{j \in NU\{d(k)\}} x_{ij}^k = 1 \quad \forall i \in N \quad (4.15)$$

$$\sum_{j \in NU\{d(k)\}} x_{o(k)j}^k = 1 \quad \forall k \in M \quad (4.16)$$

$$\sum_{i \in N \cup \{o(k)\}} x_{id(k)}^k = 1 \quad \forall k \in M \quad (4.17)$$

$$\sum_{j \in N \cup \{d(k)\}} x_{ij}^k - \sum_{j \in N \cup \{o(k)\}} x_{ji}^k = 0 \quad \forall k \in M, \forall i \in N \quad (4.18)$$

$$T_i^k + t_i^k - T_j^k \leq (1 - x_{ij}^k) M_{ij}^k \quad \forall k \in M, \forall (i, j) \in A^k \quad (4.19)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall k \in M, \forall (i, j) \in A^k \quad (4.20)$$

Nesse modelo, pode-se notar que o tempo de serviço (com seu valor de custo associado) é representado no termo 4.12. O termo 4.13 minimiza as violações nas janelas de tempo dos navios. Já o termo 4.14 minimiza as violações nas janelas de tempo dos berços.

Analisando as restrições do modelo acima, pode-se notar que se trata de um *Problema de Roteamento de Veículos com Garagens Múltiplas SEM Janelas de Tempo*, ou seja, um problema cuja resolução é menos árdua em relação ao modelo descrito na seção anterior (com janelas de tempo). Deve-se destacar que esse modelo (4.12 a 4.20) pode resultar em soluções inviáveis para o PAB, porém essas inviabilidades são eliminadas (por meio da penalizações) durante a execução dos métodos apresentados a seguir.

4.4 *Simulated Annealing* aplicado ao PAB

Inicialmente, é proposta uma versão do SA (ver Seção 2.5) para resolver o modelo apresentado na seção anterior. Essa versão é baseada no SA apresentado para o DARP (ver Seção 3.4).

4.4.1 Geração da solução inicial

A solução inicial é gerada por duas heurísticas: *heurística de distribuição para o PAB* e *heurística de programação para o PAB*. A heurística de distribuição é responsável pela atribuição dos navios aos berços. Essa heurística é baseada na heurística de distribuição para o DARP (Figura 3.2) e na heurística *FCFS-G*, apresentada em Cordeau et al. (2005). Já a heurística de programação determina o horário de atendimento dos navios nos berços.

Na heurística de distribuição (Figura 4.3) são criados inicialmente m berços vazios. Os n navios são organizados por ordem de chegada ao porto, e são distribuídos

sequencialmente aos berços de forma aleatória, porém sempre verificando se o berço selecionado poderá atender o navio em questão. Após essa distribuição, fica garantido que cada navio foi atribuído a um berço que poderá atendê-lo. O horário em que o navio será atendido ainda poderá ser incoerente, podendo apresentar sobreposição e/ou violações nas janelas de tempo, tanto dos navios quanto do berço.

1. CRIAR (m berços vazios);
2. CRIAR (uma lista L com todos os navios);
3. ORDENAR (a lista L pelo horário de chegada dos navios ao porto);
4. PARA (cada navio j em L, j = 1, 2, ..., n) FAÇA
5. SELECIONAR (um berço i, i = 1, 2, ..., m);
6. SE (o berço i não puder atender ao navio j)
7. VOLTAR (para o passo 5);
8. SENÃO
9. ATRIBUIR (o navio j ao berço i);
10. FIM-PARA;

Figura 4.3 - Heurística de distribuição para o PAB.

Na heurística de programação (Figura 4.4), são efetuados os cálculos do horário de atracação de cada navio em um berço. Além disso, a função objetivo também é calculada para cada berço k, e o somatório das funções objetivos (4.12, 4.13 e 4.14) de todos os berços resulta na função objetivo da solução. Nessa heurística, a sobreposição de horários é eliminada pelo cálculo do horário de atracação dos navios.

1. SEJA (k um berço qualquer);
2. PARA (cada navio i atribuído a k) FAÇA
3.
$$T_i^k = \begin{cases} \max(a_i, s^k), & i = 1 \\ \max(a_i, T_{i-1}^k + t_{i-1}^k), & i > 1 \end{cases}$$
4. FIM-PARA;
5. CALCULAR (a função objetivo [4.12, 4.13 e 4.14] para o berço k);

Figura 4.4 - Heurística de programação para o PAB.

4.4.2 Estrutura de vizinhança

De forma análoga à apresentada para o DARP, foram utilizados três diferentes movimentos de troca: *re-ordenar navios*, *re-alocar navio* e *trocar navios*. Assim como na geração da solução inicial, esses movimentos garantem que cada navio seja atribuído apenas a berços que possam atendê-los. Esses movimentos são baseados naqueles propostos para o DARP (ver Seção 3.4), entretanto, após a execução de cada um

desses movimentos, a heurística de programação é aplicada a cada berço para eliminar as sobreposições e recalcular o valor da função objetivo da nova solução.

O movimento *re-ordenar navios* (Figura 4.5) consiste basicamente em selecionar um berço qualquer, selecionar um navio qualquer atendido por esse berço (a), selecionar uma nova posição na sequência de atendimento desse berço (b) e trocar a posição de atendimento do navio selecionado (c).

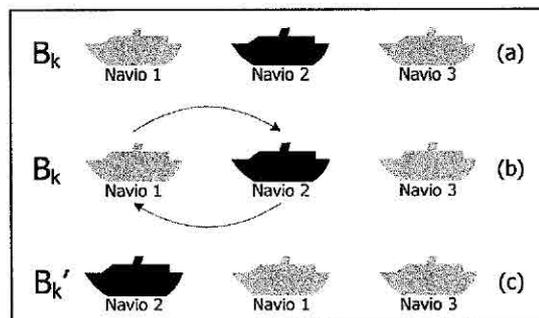


Figura 4.5 - Movimento re-ordenar navios.

Já o movimento *re-alocar navio* (Figura 4.6) consiste basicamente em selecionar dois berços quaisquer, selecionar um navio qualquer em apenas um dos dois berços (a), extraí-lo de seu berço atual e atribuí-lo ao outro berço (b). O novo berço onde o navio será atribuído deverá obrigatoriamente poder atender ao navio selecionado, pois caso contrário, outro berço deverá ser selecionado. Após a atribuição, a sequência de atendimento do novo berço deverá ser reorganizada pelo horário de chegada dos navios (c).

O movimento *trocar navios* consiste basicamente em selecionar dois berços quaisquer, selecionar um navio qualquer em cada um dos dois berços (a), e trocá-los (b). Caso os navios não possam ser atendidos pelos “novos” berços onde serão alocados, deverão ser selecionados novos navios e/ou berços. Após a troca, a sequência de atendimento dos dois berços deverá ser ordenada pelo horário de chegada dos navios (c). Esse movimento é ilustrado na Figura 4.7.

A partir dessa estrutura de vizinhança, o SA é implementado de uma forma em que cada solução vizinha é gerada por apenas um desses movimentos, sendo a sua escolha feita de forma aleatória, porém uniformemente distribuída, possibilitando assim uma

boa diversidade entre as soluções intermediárias geradas, e conseqüentemente uma boa exploração do espaço de soluções.

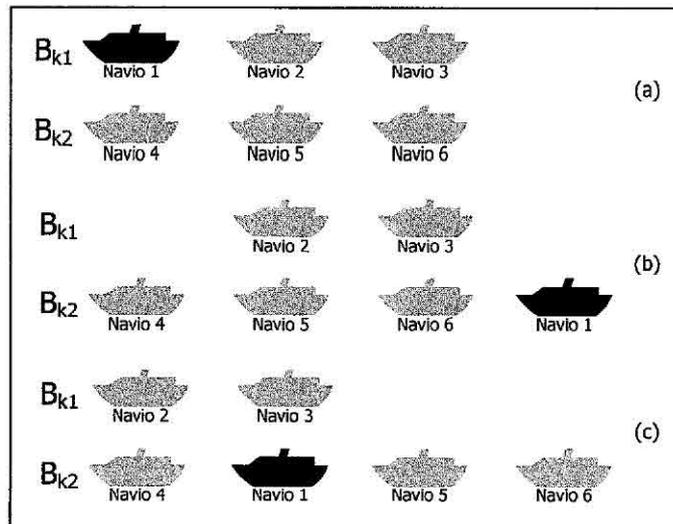


Figura 4.6 - Movimento re-alocar navios.

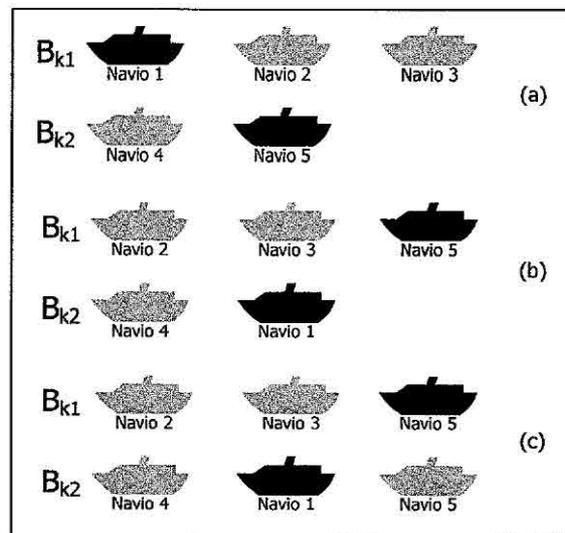


Figura 4.7 - Movimento trocar navios.

A função objetivo $f(S)$ utilizada para avaliar as soluções é aquela descrita pelas expressões 4.12, 4.13 e 4.14, e as restrições do modelo (4.15 a 4.20) são atendidas implicitamente nas heurísticas de distribuição e programação e nos movimentos de troca. O SA implementado é análogo ao proposto para o DARP (ver Figura 3.7).

4.5 ATP aplicado ao PAB

Além do SA descrito na seção anterior, é proposta também uma versão do ATP (ver Subseção 2.6.2) para resolver o modelo apresentado na Seção 4.3. Nesse método, cada indivíduo representa uma possível solução para o problema, e é descrito como uma matriz na qual cada linha representa um berço, e as colunas a sequencia de atendimento dos navios. Os elementos dessa matriz indicam o navio que deverá ser atendido. Para simplificar a implementação, cada linha da matriz deverá apresentar um número diferente de colunas (número de navios atendidos pelo respectivo berço), e todos os navios deverão ser distribuídos nessa matriz. A Figura 4.8 apresenta um possível indivíduo para um problema com 4 berços (linhas) e 14 navios.

4	10	6	13	
2	9			
7	1	3	8	12
11	14	5		

Figura 4.8 - Representação de um indivíduo no ATP.

A população inicial Pop é gerada pela heurística de distribuição para o PAB (Figura 4.3). Essa heurística é executada por Φ (tamanho da população inicial) vezes, gerando assim Φ indivíduos (possíveis soluções para o PAB). Na sequencia, a heurística de programação para o PAB (Figura 4.4) é aplicada a cada berço (linha) de cada indivíduo i ($i=1, \dots, \Phi$).

O cruzamento implementado é apresentado na Figura 4.9. Nesse operador, os indivíduos (soluções) base e guia (S_1 e S_2 , respectivamente) são “clonados”, gerando assim dois novos indivíduos S'_1 e S'_2 . A partir de então, um navio é selecionado, aleatoriamente, em um berço qualquer (nesse exemplo o berço B_2) do indivíduo base clonado S'_1 . Logo, o berço em que esse navio está alocado na solução guia clonada S'_2 é identificado, no exemplo dessa figura é o berço B_3 . Em seguida, o navio selecionado (nesse exemplo o navio 4) é retirado de seu berço de origem e alocado ao berço em que esse navio se encontra no outro indivíduo, ou seja, o navio 4 é retirado de B_2 em S'_1 e alocado em B_3 , também em S'_1 . Já em S'_2 , o navio 4 é retirado de B_3 e

alocado a B_2 . Feito isso, a sequencia de atendimento nos berços alterados B_2 e B_3 , em ambas as soluções S'_1 e S'_2 , é re-ordenada pelo horário de chegada dos navios. Por fim, a heurística de programação para o PAB (Figura 4.4) é aplicada a cada berço nos indivíduos S'_1 e S'_2 . É interessante destacar que, na forma como esse operador é implementado, o navio selecionado sempre poderá ser atendido pelos berços selecionados. Ao contrário do ATP tradicional (ver Subseção 2.6.2), esse cruzamento produz dois novos indivíduos, e não apenas um.

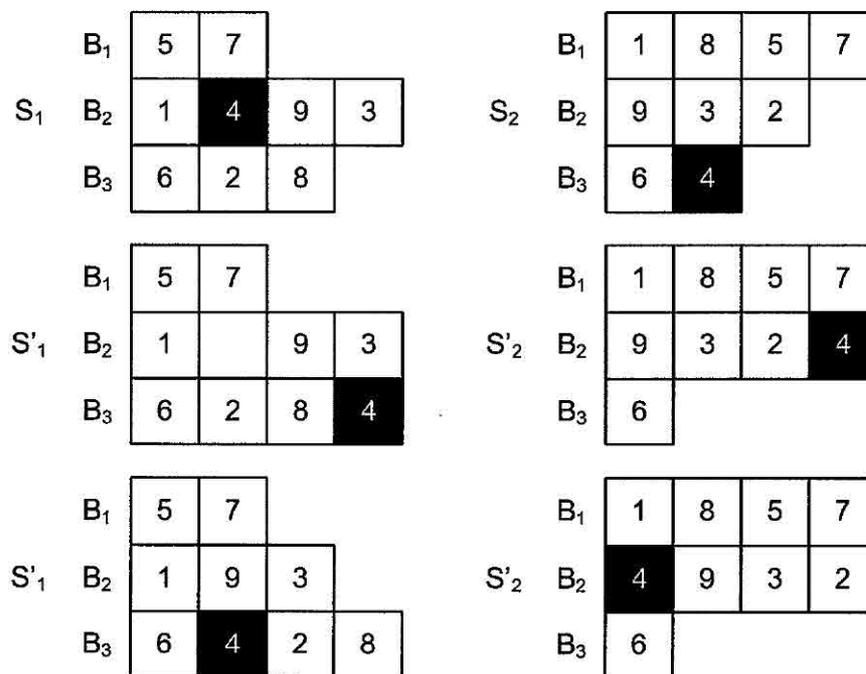


Figura 4.9 - Cruzamento utilizado no ATP.

A mutação implementada é simples, e consiste em aplicar um dos movimentos de troca utilizados pelo o SA (ver Subseção 4.4.2) para cada um dos dois indivíduos gerados. Como apresentado na Subseção 2.6.2, a mutação é executada com probabilidade τ . Assim como no SA apresentado na seção anterior, a escolha do movimento a ser executado é definida de forma aleatória, porém uniformemente distribuída.

A heurística de treinamento (determinação da função h) utilizada é baseada na aplicação da mutação por ψ (tamanho da vizinhança) vezes, sendo a melhor função g obtida, por todas as mutações, considerada como a função h do indivíduo em questão.

Assim como proposto para o SA (ver Seção 4.4), a função objetivo $f(S)$ utilizada para avaliar os indivíduos é aquela descrita pelas expressões 4.12, 4.13 e 4.14. O ATP proposto é análogo ao original (Figura 2.2), porém considera a geração de dois indivíduos filhos a cada recombinação (cruzamento), e conseqüentemente a mutação é aplicada, com probabilidade τ , a esses dois filhos gerados.

4.6 ATP/PL aplicado ao PAB

Um dos pontos principais do ATP/PL (ver Seção 2.7) é a formação de um Problema de Particionamento de Conjuntos (PPC) pelas colunas geradas durante o processo (2.19 a 2.21). Assim, para sua aplicação ao PAB, pode-se definir esse PPC onde as colunas representam a programação para os berços e as linhas os navios. O custo c_j de cada coluna j é referente ao custo do respectivo berço (Equação 4.22), e os coeficientes a_{ij} da matriz de restrições indicam se a coluna (berço) j atende o navio (linha) i . As variáveis de decisão x_j indicam se a coluna j irá pertencer à solução do problema ou não.

No caso específico do PAB, ainda deve-se considerar que cada berço possui suas próprias características, podendo não ter a capacidade de atender a um determinado navio. Nesse caso, deve-se garantir que seja utilizado nenhum ou apenas um berço de cada “tipo” disponível no cais, ou seja, cada coluna pertencente à solução final do problema deverá representar um berço distinto, e sem repetições. Para isso, deve-se inserir uma nova restrição no PPC (4.21), formando assim um problema de particionamento de conjuntos com uma restrição adicional (PPC^+).

$$\sum_{j=1}^p b_{ij}x_j \leq 1 \quad i = 1, \dots, m \quad (4.21)$$

Cada elemento $b_{ij} \in \{0,1\}$, sendo $i \in M = \{1, \dots, m\}$ e $j \in P = \{1, \dots, p\}$. m é o número de berços disponíveis e p o número de colunas geradas. $b_{ij} = 1$ se a coluna j representar o berço i . Cada coluna é representada por um “indivíduo” (Figura 4.10) formado por um vetor de números inteiros, onde a primeira posição indica o berço referente à coluna, e as demais posições representam os navios atendidos por esse berço (coluna).

O indivíduo representado na Figura 4.10 indica que o berço k atenderá os navios i , n e $i+1$, nessa ordem. É interessante notar que no ATP/PL cada indivíduo representará um berço (coluna), diferentemente do ATP descrito na seção anterior, no qual cada

indivíduo representa uma solução (conjunto de m berços) para o problema original.

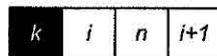


Figura 4.10 - Representação de um indivíduo no ATP/PL.

O custo de cada coluna (indivíduo) é calculado de forma análoga à apresentada para o SA e o ATP (ver Seções 4.4 e 4.5), porém, para facilitar a compreensão deste método, pode-se considerar B^k como sendo a sequência de atendimento do berço k , e assim o custo de cada coluna é apresentado na Equação 4.22.

$$c_k = \begin{cases} \omega_0 \sum_{i \in B^k} v_i (T_i^k - a_i + t_i^k) \\ + \omega_1 \sum_{i \in B^k} (\max(0, a_i - T_i^k) + \max(0, T_i^k + t_i^k - b_i)) \\ + \omega_2 (\max(0, s^k - T_{o(k)}^k) + \max(0, T_{d(k)}^k + e^k)) \end{cases} \quad (4.22)$$

O conjunto inicial de colunas é gerado aleatoriamente, porém de forma direcionada ao problema, ou seja, contendo colunas que formem uma solução para o PPC^+ . Essas colunas são geradas por uma única execução da heurística de distribuição para o PAB (Figura 4.3) seguida pela execução da heurística de programação para o PAB (Figura 4.4) para cada coluna. É interessante destacar que a solução formada por essas colunas provavelmente será inválida, pois as colunas podem apresentar violações nas janelas de tempo. Porém, devido ao seu alto custo (devido às penalizações), essas colunas deixarão de pertencer à solução à medida que novas colunas forem criadas.

A partir do conjunto inicial de colunas gerado é formado o PMR (PPC^+) inicial, cuja relaxação linear é resolvida pela PL. Feito isso, novas colunas são geradas pelo ATP (subproblema *pricing*), considerando os valores das variáveis duais para construir as funções de adaptação g e h . As colunas válidas (que não apresentam violações nas janelas de tempo) e com custo reduzido negativo são adicionadas ao atual PPC^+ , e este é resolvido novamente pela PL. Esse processo é repetido por um determinado número de iterações ou enquanto um número máximo de colunas não for gerado. Finalmente, terminada essa interação, o PPC^+ é resolvido de forma inteira (PLI) por meio do *software* CPLEX 10.0 (ILOG, 2006). Vale destacar que

caso não seja gerado um conjunto suficiente de colunas, esse problema poderá ser infactível. Assim, uma boa prática para evitar que isso aconteça é a formação do PMR inicial com colunas que representem uma solução factível para o PAB.

A interação do ATP com a PL é baseada principalmente na definição da função de adaptação (função g) dos indivíduos no ATP. Essa função é definida com o uso de informações das variáveis duais retiradas da PL. Para o PAB, a função g é definida da seguinte maneira:

$$g_j = \frac{c_j}{\sum_{i=1}^n \pi_i a_{ij} + \sum_{i=n+1}^{n+m} \pi_i b_{ij}} \quad \text{para} \quad \left(\sum_{i=1}^n \pi_i a_{ij} + \sum_{i=n+1}^{n+m} \pi_i b_{ij} > 0 \right) \quad (4.23)$$

$$g_j = c_j \quad \text{para} \quad \left(\sum_{i=1}^n \pi_i a_{ij} + \sum_{i=n+1}^{n+m} \pi_i b_{ij} \leq 0 \right) \quad (4.24)$$

c_j é o custo da coluna j (Equação 4.22) e π_i é a variável dual para a para a restrição i . O custo reduzido da j -ésima coluna inserida no PPC^+ (PMR), denotado por θ_j , é calculado da seguinte maneira:

$$\theta_j = c_j - \left(\sum_{i=1}^n \pi_i a_{ij} + \sum_{i=n+1}^{n+m} \pi_i b_{ij} \right) \quad (4.25)$$

A cada iteração do ATP/PL, a população inicial Pop do ATP é gerada de forma análoga à apresentada na Seção 4.5, ou seja, pela heurística de distribuição para o PAB (Figura 4.3) e da heurística de programação para o PAB (Figura 4.3). A heurística de distribuição é executada por Φ (tamanho da população inicial) vezes.

Após a execução da heurística de programação, que irá calcular o custo “real” c_j (Equação 4.22) dos indivíduos, são calculados os valores das funções de aptidão g e h e do $rank$ do indivíduo j . A função g é calculada como apresentado nas equações 4.23 e 4.24. O $rank$ é calculado de forma análoga à apresentada no ATP e no ATP/PL original (Equação 2.17). Já a função h é obtida pela heurística de treinamento apresentada na Figura 4.11. Essa heurística é baseada em uma busca local simples, que é responsável por avaliar vários indivíduos (colunas) alternativos em uma vizinhança.

```

1. SEJA (k uma coluna qualquer);
2.  $k' \leftarrow k$ ;
3.  $h^* = g(k')$ 
4. POR ( $\psi$  vezes)
5.    $i \leftarrow$  um navio qualquer atendido pela coluna  $k'$ ;
6.    $j \leftarrow$  outro navio qualquer atendido pela coluna  $k'$ ;
7.   TROCAR (a ordem de atendimento dos navios i e j);
8.   EXECUTAR (a heurística de programação para a coluna  $k'$ );
9.   CALCULAR (a função g para a coluna  $k' - g(k')$ );
10.    SE ( $g(k') < h^*$ );
11.       $h^* \leftarrow g(k')$ ;
12.    FIM-SE;
13. FIM-POR;
14.  $h(k) \leftarrow h^*$ ;

```

Figura 4.11 - Heurística de treinamento utilizada no ATP/PL.

A mutação utilizada também é baseada em uma busca local, implementada por meio de uma simples troca das posições de atendimento de dois navios (selecionados aleatoriamente) atendidos por uma coluna (indivíduo). Esse processo é descrito na Figura 4.12.

```

1. SEJA (k uma coluna qualquer);
2.  $i \leftarrow$  um navio qualquer atendido pela coluna k;
3.  $j \leftarrow$  outro navio qualquer atendido pela coluna k;
4. TROCAR (a ordem de atendimento dos navios i e j);
5. EXECUTAR (a heurística de programação para a coluna k);
6. CALCULAR (a função  $g(k)$ );
7. EXECUTAR (a heurística de treinamento - obter  $h(k)$ );
8. CALCULAR (o rank  $\delta(k)$ );

```

Figura 4.12 - Mutação utilizada no ATP/PL.

O cruzamento é responsável por gerar novos indivíduos da seguinte maneira: dois indivíduos são selecionados (base e guia). A partir de então, é criado um novo indivíduo semelhante ao base. Assim, cada navio atendido pelo indivíduo guia é inserido nesse novo indivíduo, caso o berço correspondente ao novo possa atender ao navio em questão. Feito isso, a sequência de atendimento do novo indivíduo é ordenada pelo horário de chegada dos navios ao porto. O cruzamento é descrito na Figura 4.13.

Esses operadores são incorporados ao ATP, cujo pseudocódigo é similar ao original, apresentado na Figura 2.2 (ver Subseção 2.6.2). É interessante notar que, utilizando esses processos, o ATP formará populações de vários tamanhos, guiados pelo objetivo de selecionar colunas de baixo custo e com uma cobertura suficiente dos navios.

As melhores colunas deverão incluir um número de navios diversificado, caracterizado pela heurística de treinamento (função de treinamento) que guiará o processo evolutivo.

```

1. SEJA (k uma coluna base);
2. SEJA (k' uma coluna guia);
3.  $k'' \leftarrow \text{clone}(k)$ ;
4. PARA (cada navio i atendido pela coluna k') FAÇA
5.   SE (o berço referente à coluna k'' for capaz de atender ao navio i)
6.     INSERIR (o navio i na coluna k'');
7.   FIM-SE;
8. FIM-PARA;
9. ORDENAR (a sequência de atendimento da coluna k'');
10. EXECUTAR (a heurística de programação para a coluna k'');
11. CALCULAR (a função  $g(k)$ );
12. EXECUTAR (a heurística de treinamento - obter  $h(k)$ );
13. CALCULAR (o rank  $\delta(k)$ );
14. INSERIR (a coluna k'' na população);

```

Figura 4.13 - Operador de cruzamento utilizado no ATP/PL.

4.7 Experimentos computacionais

Para avaliar o desempenho dos métodos propostos neste capítulo, foram utilizadas 30 instâncias distintas, cada uma com 60 navios e 13 berços. Essas instâncias foram geradas aleatoriamente, e apresentadas em Cordeau et al. (2005).

Com o intuito de “melhorar” as soluções obtidas pelo SA, foi aplicado um “re-aquecimento” (SA+RA). Essa técnica consiste em, após executar o SA, aplicá-lo novamente à melhor solução obtida até então como solução inicial. No re-aquecimento, foram utilizados diferentes valores de parâmetros. A temperatura inicial foi reduzida em relação ao SA “normal”, e o número máximo de iterações foi aumentado. Dessa forma, a busca por melhores soluções é intensificada na região do espaço de busca próxima à solução inicial, ou seja, o re-aquecimento faz um refinamento na solução obtida pelo SA.

Os parâmetros utilizados no SA foram: $\alpha = 0,975$, $T_i = 40000$, $T_c = 0,01$ e $SA_{max} = 1000$. Já no re-aquecimento, foram utilizados os seguintes valores: $\alpha = 0,975$, $T_i = 10000$, $T_c = 0,01$ e $SA_{max} = 2000$. Os resultados obtidos pelo SA, com e sem o re-aquecimento, são apresentados na Tabela 4.1. Nessa tabela, pode-se observar que as soluções obtidas com o re-aquecimento foram em média 0,03% melhores em relação às obtidas pelo SA, ou seja, o re-aquecimento proporcionou apenas uma pequena melhora nas soluções (para 5 instâncias apenas).

Tabela 4.1 - Resultados obtidos pelo SA para o PAB.

Instância	SA		SA+RA		Melhora (%)
	$f(S)$	Tempo (seg.)	$f(S)$	Tempo (seg.)	
i01	1409	19,44	1409	53,12	0
i02	1261	20,17	1261	58,94	0
i03	1129	19,77	1129	54,03	0
i04	1302	21,03	1302	67,33	0
i05	1207	20,00	1207	55,38	0
i06	1261	19,77	1261	53,88	0
i07	1279	20,69	1279	60,52	0
i08	1299	25,95	1299	61,45	0
i09	1444	22,20	1444	57,91	0
i10	1213	25,33	1213	68,95	0
i11	1368	27,88	1368	76,77	0
i12	1325	22,61	1325	62,84	0
i13	1360	24,94	1360	68,19	0
i14	1233	25,94	1233	75,06	0
i15	1295	20,69	1295	54,55	0
i16	1366	20,77	1364	63,91	0,15
i17	1283	20,08	1283	56,28	0
i18	1345	19,70	1345	53,98	0
i19	1370	19,22	1370	52,83	0
i20	1329	19,52	1328	53,38	0,08
i21	1344	19,56	1341	53,52	0,22
i22	1326	21,23	1326	57,97	0
i23	1266	19,66	1266	53,75	0
i24	1260	19,75	1260	54,09	0
i25	1377	19,52	1377	53,56	0
i26	1319	20,78	1318	57,34	0,08
i27	1261	23,64	1261	69,98	0
i28	1360	21,42	1360	58,47	0
i29	1280	25,61	1280	69,09	0
i30	1349	26,86	1344	70,67	0,37
Média	1307,33	21,79	1306,93	60,26	0,03

Para o ATP, os parâmetros foram $\Phi = 500$, $Step = 0,000000001$, $ATP_{max} = 100000$, $\kappa = 10$, $\tau = 60$ e $\psi = 10$. O valor de g_{max} utilizado foi obtido a partir da maior função g dos indivíduos gerados na população inicial. Já o valor de β , inicialmente, foi igual a 0, e $\gamma = 1/g_{max}$. Os resultados obtidos pelo ATP são apresentados na Tabela 4.2.

Tabela 4.2 - Resultados obtidos pelo ATP para o PAB.

Instância	$f(S)$	Tempo	Instância	$f(S)$	Tempo
i01	1415	142,69	i16	1367	146,19
i02	1261	140,02	i17	1291	142,73
i03	1146	146,77	i18	1345	140,72
i04	1303	143,80	i19	1378	143,39
i05	1207	149,80	i20	1335	144,38
i06	1269	142,70	i21	1343	144,77
i07	1284	144,92	i22	1341	145,31
i08	1310	146,19	i23	1266	139,88
i09	1444	147,36	i24	1261	144,83
i10	1213	141,73	i25	1382	144,42
i11	1387	147,20	i26	1326	150,26
i12	1327	145,08	i27	1261	148,97
i13	1361	143,52	i28	1365	145,81
i14	1233	144,64	i29	1293	143,50
i15	1296	143,59	i30	1368	148,39
$f(S)$ média:	1312,60		Tempo médio:	144,79	

No ATP/PL, os parâmetros utilizados foram $\Phi = 10$, $Step = 0.001$, $ATP_{max} = 70$, $\kappa = 10$, $\tau = 60$, $\psi = 6$ e $\gamma = 0,01$. O valor de g_{max} utilizado foi obtido a partir da maior função g dos indivíduos gerados na população inicial. Já o valor de β , inicialmente, foi igual a 0. O número de colunas geradas foi limitado em 70.000, e o número de iterações em 10.000. Os resultados obtidos pelo ATP/PL são apresentados na Tabela 4.3.

Na Tabela 4.3 pode-se notar que o número de colunas geradas para cada instância é relativamente baixo em relação ao número máximo de colunas previsto (70.000). Pode-se notar também que o valor da solução do último PPC^+ (formado por todas as colunas geradas) foi o mesmo em todos os casos, sendo resolvido tanto pela PL quanto pela PLI, o que indica a obtenção da solução ótima para o PPC^+ formado pelas colunas geradas. O tempo para geração das colunas (interação entre o ATP e a PL) também é baixo, assim como o tempo para resolução do PPC^+ final pela PLI, o que resulta em um tempo total de processamento competitivo para o ATP/PL.

Tabela 4.3 - Resultados obtidos pelo ATP/PL para o PAB.

Inst.	Colunas geradas	PPC ⁺ (via PL)	PPC ⁺ (via PLI)	Tempo (seg.)		
				GC	PLI	Total
i01	26664	1409,00	1409	72,14	2,47	74,61
i02	12752	1261,00	1261	58,92	1,83	60,75
i03	70000	1129,00	1129	94,62	40,83	135,45
i04	54612	1302,00	1302	103,16	7,02	110,17
i05	70019	1207,00	1207	72,20	52,50	124,70
i06	25990	1261,00	1261	74,22	4,12	78,34
i07	70023	1279,00	1279	86,73	27,47	114,20
i08	70005	1299,00	1299	48,77	8,30	57,06
i09	37846	1444,00	1444	91,86	4,61	96,47
i10	70005	1213,00	1213	61,81	37,59	99,41
i11	43507	1369,00	1369	95,34	4,00	99,34
i12	18508	1325,00	1325	77,39	3,30	80,69
i13	70017	1360,00	1360	62,55	27,39	89,94
i14	26221	1233,00	1233	69,05	4,91	73,95
i15	70002	1295,00	1295	71,28	2,91	74,19
i16	30063	1365,00	1365	169,81	0,55	170,36
i17	70033	1283,00	1283	32,89	13,67	46,58
i18	36108	1345,00	1345	81,78	2,23	84,02
i19	16135	1367,00	1367	122,00	1,19	123,19
i20	20528	1328,00	1328	74,25	8,05	82,30
i21	48386	1341,00	1341	103,52	4,56	108,08
i22	54140	1326,00	1326	104,17	1,20	105,38
i23	70010	1266,00	1266	41,59	2,12	43,72
i24	70008	1260,00	1260	75,81	3,09	78,91
i25	41210	1376,00	1376	95,09	1,48	96,58
i26	70011	1318,00	1318	70,00	31,11	101,11
i27	37022	1261,00	1261	77,38	5,48	82,86
i28	70004	1360,00	1360	51,52	1,39	52,91
i29	70001	1280,00	1280	196,36	7,00	203,36
i30	7837	1344,00	1344	69,62	1,39	71,02
Média	48256	1306,87	1306,87	83,53	10,46	93,99

A Tabela 4.4 apresenta a melhora (%) obtida entre os resultados das soluções encontradas pelas abordagens propostas. Nessa tabela, percebe-se claramente a competitividade entre os métodos propostos, pois a melhora é expressivamente pequena entre todos os métodos. Entretanto, percebe-se também que tanto o SA+RA quanto o ATP/PL apresentam melhoras em relação ao ATP, e entre si (SA+RA × ATP/PL), os resultados são praticamente idênticos. Além disso, os tempos de processamento foram satisfatórios e próximos entre os métodos propostos.

Tabela 4.4 - Comparação entre os métodos propostos para o PAB.

Inst.	SA+RA	ATP/PL	ATP/PL	Inst.	SA+RA	ATP/PL	ATP/PL
	× ATP	× ATP	× SA+RA		× ATP	× ATP	× SA+RA
i01	0,43	0,43	0,00	i16	0,22	0,15	-0,07
i02	0,00	0,00	0,00	i17	0,62	0,62	0,00
i03	1,51	1,51	0,00	i18	0,00	0,00	0,00
i04	0,08	0,08	0,00	i19	0,58	0,80	0,22
i05	0,00	0,00	0,00	i20	0,53	0,53	0,00
i06	0,63	0,63	0,00	i21	0,15	0,15	0,00
i07	0,39	0,39	0,00	i22	1,13	1,13	0,00
i08	0,85	0,85	0,00	i23	0,00	0,00	0,00
i09	0,00	0,00	0,00	i24	0,08	0,08	0,00
i10	0,00	0,00	0,00	i25	0,36	0,44	0,07
i11	1,39	1,31	-0,07	i26	0,61	0,61	0,00
i12	0,15	0,15	0,00	i27	0,00	0,00	0,00
i13	0,07	0,07	0,00	i28	0,37	0,37	0,00
i14	0,00	0,00	0,00	i29	1,02	1,02	0,00
i15	0,08	0,08	0,00	i30	1,79	1,79	0,00
	SA+RA × ATP				ATP/PL × ATP		ATP/PL × SA+RA
Média:	0,43%				0,44 %		0,00 %

Vale destacar ainda que, em todos os experimentos realizados, as penalizações utilizadas para todos os métodos propostos foram $\omega = [1,10,10]$. Além disso, todas as soluções obtidas foram viáveis, ou seja, satisfazem o modelo original do problema (4.1 a 4.11). Os métodos propostos foram executados por cinco vezes, com diferentes sementes, e o desvio obtido (calculado de forma similar à apresentada na Equação 3.66) foi de 0,04%, 1,88% e 0,05%, para o SA+RA, ATP e ATP/PL, respectivamente.

As melhores soluções obtidas pelos métodos propostos ainda foram comparadas com as melhores soluções conhecidas para as instâncias utilizadas. Essas melhores soluções foram obtidas por uma heurística baseada na Busca Tabu (BT), apresentada em Cordeau et al. (2005). Além disso, o CPLEX 10.0 (ILOG, 2006) também foi utilizado, de forma isolada, para resolver o modelo descrito na Seção 4.2. Foi utilizado um limite máximo de processamento de 1 hora para cada instância. A Tabela 4.5 apresenta essas comparações.

Como pode ser observado na Tabela 4.5, as soluções obtidas pelo SA+RA e pelo ATP/PL foram pouco melhores do que às obtidas pela Busca Tabu proposta por Cordeau et al. (2005). A Busca Tabu obteve apenas uma solução melhor do que o

SA+RA e o ATP/PL (instância i10), e várias soluções melhores do que as apresentadas pelo ATP. Já o CPLEX não foi capaz de obter soluções para várias instâncias (em 1 hora), e nos casos em que foram encontradas soluções, os resultados foram expressivamente piores do que todos os métodos propostos (> 169%).

Tabela 4.5 - Comparação com a literatura e com o CPLEX.

Inst.	Busca	CPLEX		Melhoras (%)					
	Tabu			SA+RA		ATP		ATP/PL	
	$f(S)$	$f(S)$	Gap	BT	CPX	BT	CPX	BT	CPX
i01	1415	-	-	0,43	-	0,00	-	0,43	-
i02	1263	2606	3,82	0,16	106,66	0,16	106,66	0,16	106,66
i03	1139	2565	4,00	0,89	127,19	-0,61	123,82	0,89	127,19
i04	1303	4353	8,62	0,08	234,33	0,00	234,08	0,08	234,33
i05	1208	2672	4,89	0,08	121,38	0,08	121,38	0,08	121,38
i06	1262	-	-	0,08	-	-0,55	-	0,08	-
i07	1279	2887	4,73	0	125,72	-0,39	124,84	0	125,72
i08	1299	5177	11,69	0	298,54	-0,84	295,19	0	298,54
i09	1444	-	-	0	-	0,00	-	0	-
i10	1212	-	-	-0,08	-	-0,08	-	-0,08	-
i11	1378	-	-	0,73	-	-0,65	-	0,66	-
i12	1325	3206	5,48	0	141,96	-0,15	141,60	0	141,96
i13	1360	-	-	0	-	-0,07	-	0	-
i14	1233	-	-	0	-	0,00	-	0	-
i15	1295	4672	9,77	0	260,77	-0,08	260,49	0	260,77
i16	1375	4320	8,97	0,81	216,72	0,59	216,02	0,73	216,48
i17	1283	-	-	0	-	-0,62	-	0	-
i18	1346	3681	6,94	0,07	173,68	0,07	173,68	0,07	173,68
i19	1370	2400	3,04	0	75,18	-0,58	74,17	0,22	75,57
i20	1328	-	-	0	-	-0,52	-	0	-
i21	1346	-	-	0,37	-	0,22	-	0,37	-
i22	1332	3489	7,31	0,45	163,12	-0,67	160,18	0,45	163,12
i23	1266	-	-	0	-	0,00	-	0	-
i24	1261	4867	10,13	0,08	286,27	0,00	285,96	0,08	286,27
i25	1379	1993	2,67	0,15	44,73	-0,22	44,21	0,22	44,84
i26	1330	2520	3,62	0,91	91,20	0,30	90,05	0,91	91,20
i27	1261	3209	5,70	0	154,48	0,00	154,48	0	154,48
i28	1365	-	-	0,37	-	0,00	-	0,37	-
i29	1282	4809	9,43	0,16	275,70	-0,85	271,93	0,16	275,70
i30	1351	-	-	0,52	-	-1,24	-	0,52	-
Méd.	1309,67	3495,65	6,52	0,21	170,45	-0,22	169,34	0,21	170,46

Em relação ao tempo para obtenção das soluções, o CPLEX utilizou 1 hora para cada instância (3600 seg.). A Busca Tabu utilizou aproximadamente 120 segundos para cada instância, segundo descrito em Cordeau et al. (2005). Já o SA+RA, o ATP, e o ATP/PL utilizaram um tempo médio de 60,26, 144,79 e 93,99 segundos, respectivamente (ver Tabelas 4.1, 4.2 e 4.3). Esses números demonstram a competitividade dos métodos propostos em relação à Busca Tabu e a superioridade em relação ao CPLEX.

4.7.1 Soluções iniciais para o CPLEX

Com intuito de verificar a “qualidade” dos resultados obtidos, as melhores soluções encontradas pelos métodos propostos (Ω) foram inseridas no CPLEX como soluções iniciais para o modelo descrito na Seção 4.2. Essa abordagem é realizada por meio da biblioteca do *software* CPLEX (ILOG, 2006), e relatos de sua aplicação não foram encontrados na literatura. A Tabela 4.6 apresenta os resultados obtidos.

Tabela 4.6 - Melhores soluções utilizadas como entrada para o CPLEX.

Inst.	Ω	CPX	Gap	Tempo	Δ	Inst.	Ω	CPX	Gap	Tempo	Δ
i01	1409	1409	0,14	3600	0	i16	1364	1364	0,19	3600	0
i02	1261	1261	0,01	42,91	0	i17	1283	1283	0,28	3600	0
i03	1129	1129	0,09	3600	0	i18	1345	1345	0,24	3600	0
i04	1302	1302	0,23	3600	0	i19	1367	1367	0,04	3600	0
i05	1207	1207	0,11	3600	0	i20	1328	1328	0,20	3600	0
i06	1261	1261	0,17	3600	0	i21	1341	1341	0,14	3600	0
i07	1279	1279	0,17	3600	0	i22	1326	1326	0,26	3600	0
i08	1299	1299	0,12	3600	0	i23	1266	1266	0,03	3600	0
i09	1444	1444	0,54	3600	0	i24	1260	1260	0,15	3600	0
i10	1213	1213	0,21	3600	0	i25	1376	1376	0,43	3600	0
i11	1368	1368	0,36	3600	0	i26	1318	1318	0,14	3600	0
i12	1325	1325	0,14	3600	0	i27	1261	1261	0,01	1927	0
i13	1360	1360	0,16	3600	0	i28	1360	1360	0,25	3600	0
i14	1233	1233	0,10	3600	0	i29	1280	1280	0,03	3600	0
i15	1295	1295	0,07	3600	0	i30	1344	1344	0,15	3600	0
Gap médio: 0,17 %						Tempo médio: 3425,67 seg.					

Pode-se observar na Tabela 4.6 que o CPLEX não foi capaz de melhorar (Δ) as soluções Ω em 1 hora de processamento. Porém, é interessante destacar que, com o uso do CPLEX, pôde-se confirmar a “qualidade” das soluções obtidas, pois o *gap* médio obtido foi de 0,17%, o que indica que as soluções estão muito próximas do

ótimo global. Para as instâncias i02 e i27 as soluções ótimas foram obtidas tanto pelo SA (com e sem re-aquecimento) quanto pelo ATP e pelo ATP/PL. Já o CPLEX, sem a utilização das soluções iniciais, não encontrou as soluções ótimas para essas instâncias, apresentando *gaps* de 3,82% e 5,70%, respectivamente.

4.8 Considerações finais

Os trabalhos existentes sobre o PAB apresentam diferentes métodos de resolução e inúmeras formulações matemáticas. Entretanto, todos têm um ponto em comum, a utilização de instâncias referentes a problemas reais encontrados em portos de várias partes do mundo. Esse fato dificulta a comparação direta com os métodos já propostos, pois em quase todos os casos, as empresas que cedem os dados para essas instâncias não permitem a sua divulgação. Esse fato pode ser confirmado por uma busca na literatura (e na Internet) por instâncias do PAB. Após uma exaustiva busca por essas instâncias, não foi encontrado nenhum conjunto disponível. Logo, as instâncias utilizadas foram cedidas, por intermédio do Prof. Dr. Alexandre César Muniz de Oliveira (UFMA), pelos autores do trabalho Cordeau et al. (2005).

O modelo proposto é capaz de representar o problema de uma forma relaxada, facilitando assim sua resolução. A alocação dos navios aos berços e a programação do atendimento dos navios foram realizadas separadamente (pelas heurísticas apresentadas) para os métodos propostos. O *Simulated Annealing*, o Algoritmo de Treinamento Populacional, e o ATP/PL, integrados com as demais heurísticas apresentadas neste capítulo, foram capazes de obter, em todos os casos e com pouco tempo de processamento, soluções válidas para o problema. Além disso, os métodos propostos se mostraram extremamente eficientes, pois os *gaps* apresentados foram expressivamente baixos, indicando uma grande proximidade com as soluções ótimas para o problema.

O ATP, integrado com uma técnica tradicional de Geração de Colunas, foi capaz de resolver o subproblema gerador de colunas (*pricing*) de uma forma implícita. Isso foi feito por meio da definição das funções g e h , que utilizam informações dos valores duais. Dessa forma, esse método mostrou ser adequado para outros problemas para os quais a geração de colunas seja indicada. Além disso, vale ressaltar que essa integração do ATP com a PL por meio da Geração de Colunas (ATP/PL) é capaz de “aprimorar” o desempenho do ATP quando utilizado separadamente.

Os métodos propostos ainda foram comparados com uma abordagem recente encontrada na literatura, e que apresentava as melhores soluções até então, e a qualidade das soluções encontradas foi ligeiramente superior, apresentando melhores resultados para as instâncias utilizadas. Além disso, os resultados obtidos ainda foram comparados com o CPLEX, e nesse caso, nota-se claramente a superioridade dos métodos propostos tanto na qualidade das soluções quanto no tempo de processamento.

Enfim, a estrutura de vizinhança (movimentos de troca), os operadores genéticos, e a heurística de treinamento foram adequadas e eficientes para exploração do espaço de soluções, e os resultados demonstram o potencial das abordagens apresentadas, onde soluções de alta qualidade são obtidas, para problemas relativamente grandes e em tempos de processamento expressivamente baixos.

Todos os experimentos apresentados neste capítulo foram realizados em um *PC* com processador *AMD Athlon™ 64 3500* de 2.2 GHz e 1GB de memória RAM. Toda a implementação foi desenvolvida na linguagem C++, e em alguns casos com chamadas à biblioteca do *software* CPLEX 10.0. As soluções obtidas por Cordeau et al. (2005) foram executadas em uma estação de trabalho *Sun Workstation* de 900MHz.

O estudo apresentado neste capítulo contou com a colaboração do Prof. Dr. Alexandre César Muniz de Oliveira, da Universidade Federal do Maranhão (UFMA).

5 PROBLEMA DE PROGRAMAÇÃO QUADRÁTICA BINÁRIA IR-RESTRITA

O Problema de Programação Quadrática Binária Irrestrita (PQ) é um dos problemas clássicos da área de otimização não-linear. Esse problema também é conhecido como Problema de Programação Bivalente Quadrática Irrestrita (GULATI et al., 1984) e Problema de Programação Quadrática Zero-Um Irrestrita (CHARDAIRE; SUTTER, 1995). Por uma questão de notação, esse problema será tratado adiante como PQ.

O PQ consiste em maximizar (ou minimizar) uma função objetivo quadrática por meio da escolha de valores apropriados para as variáveis de decisão binárias (BEASLEY, 1998). O PQ é um problema *NP-Hard* (BILLIONNET; ELLOUMI, 2007) e apresenta uma grande quantidade de aplicações em diversas áreas, como por exemplo: biologia molecular (PHILLIPS; ROSEN, 1994); biologia computacional (FORRESTER; GREENBERG, 2008); neurociência (PARDALOS et al., 2004); planejamento de investimentos e análise financeira (LAUGHUNN, 1970; MCBRIDE; YORMARK, 1980); e alguns problemas do tipo CAD (KRARUP; PRUZAN, 1978). Além disso, o PQ ainda aborda inúmeros problemas modelados por grafos, como coloração de vértices, clique máximo, máximo conjunto independente, e outros (KOCHENBERGER et al., 2005; PARDALOS; RODGERS, 1992; STRIJK et al., 2000).

5.1 Revisão da literatura

Ao contrário dos problemas apresentados nos capítulos anteriores, o PQ é um problema muito difundido e amplamente estudado. Consequentemente, inúmeros trabalhos realizados segundo seus preceitos e aplicações são encontrados na literatura.

Vários métodos exatos e heurísticos têm sido propostos para resolver o PQ, entretanto, os métodos exatos existentes (BILLIONNET; SUTTER, 1994; PARDALOS; RODGERS, 1990a; PARDALOS; RODGERS, 1990b) são restritos a problemas com até 200 variáveis. Já os métodos heurísticos (BEASLEY, 1998; GLOVER et al., 1998; PARDALOS; JHA, 1992) têm apresentado bons resultados para instâncias maiores.

Uma outra abordagem interessante para resolver o PQ é a relaxação do problema para obtenção de limitantes (ADAMS; DEARING, 1994; CHARDAIRE; SUTTER, 1995). Essa abordagem possui a vantagem de definir limitantes para a solução ótima, e pode apresentar uma informação dual de boa qualidade, o que permite avaliar a proximidade da melhor solução encontrada em relação à solução ótima do problema.

Outra prática comum para resolver o PQ é a linearização do seu modelo original (ADAMS et al., 2004; GLOVER; WOOLSEY, 1974; HANSEN; MEYER, 2008), ou seja, a obtenção de um modelo linear equivalente cujas soluções são correspondentes ao modelo quadrático original. Dessa forma, o PQ é transformado em um problema linear inteiro misto, o que permite a relaxação linear de suas variáveis de decisão, e conseqüentemente a obtenção de um limitante (*roof dual*) para a solução do problema original (ADAMS; DEARING, 1994; BOROS et al., 1990; BOROS et al., 1992).

Vários métodos baseados em busca em árvores para resolver o PQ são encontrados na literatura. Gulati et al. (1984) apresentam um método de busca em árvore, baseado na enumeração de ótimos locais, que resolve problemas com até 40 variáveis. Pardalos e Rodgers (1990a) apresentam um método de busca em árvore que utiliza limitantes baseados na fixação de variáveis em cada nó da árvore. Os resultados tratam problemas com até 200 variáveis. Pardalos e Rodgers (1990b) apresentam uma versão paralelizada de um *branch-and-bound* capaz de resolver problemas com até 100 variáveis.

Billionnet e Sutter (1994) apresentam um método de busca em árvore capaz de resolver problemas com até 100 variáveis. Palubeckis (1995) propõe método heurístico baseada em uma busca em árvore que apresenta resultados para problemas com até 247 variáveis. Huang et al. (2006) descrevem diferentes estratégias para estimar limitantes inferiores para minimização do PQ. Essas estratégias são baseadas em um algoritmo *branch-and-bound* e os resultados são apresentados para problemas com até 60 variáveis.

Uma heurística baseada na Busca Tabu é proposta por Glover et al. (1998). Essa heurística resolve problemas com até 500 variáveis. Beasley (1998) apresenta uma comparação entre duas metaheurísticas para resolver o PQ. Ele utiliza a Busca Tabu e o *Simulated Annealing* para resolver problemas com até 2500 variáveis (com baixa densidade). A Busca Tabu apresenta os melhores resultados para a maioria dos problemas utilizados. Já o *Simulated Annealing* supera a Busca Tabu para os problemas com maior número de variáveis.

Glover et al. (2002) apresentam várias heurísticas baseadas no algoritmo DDT (BOROS et al., 1989) para resolução do PQ. Essas heurísticas são testadas com várias instâncias com até 2500 variáveis. Palubeckis (2004) apresenta cinco estratégias diferentes baseadas na Busca Tabu com inicializações múltiplas para resolver o PQ. São

apresentados resultados para instâncias com até 6000 variáveis, e o tempo computacional necessário para resolvê-las é inferior aos apresentado pelos demais métodos encontrados na literatura.

Adams e Dearing (1994) apresentam uma discussão sobre a obtenção de limitantes para o PQ. Eles apresentam um modelo linear para o problema, que é obtido pela técnica de linearização de Glover e Woolsey (1974). Várias estratégias de linearização do PQ são apresentadas e discutidas em Adams et al. (2004) e Hansen e Meyer (2008).

Chardaire e Sutter (1995) propõem um algoritmo para obtenção de limitantes para o PQ. Esse algoritmo é baseado na Decomposição Lagrangiana da função objetivo quadrática do problema original em um somatório de funções pseudo-lineares. O método proposto é uma relaxação dual do problema quadrático original. A divisão do problema em *clusters* é realizada de forma aleatória. Os resultados obtidos mostram que quanto maior o tamanho dos *clusters* melhor é o limitante, entretanto, os *gaps* aumentam significativamente com o tamanho dos problemas. Eles mostram que entre as técnicas de decomposição, a lagrangiana é a que apresenta os melhores resultados. Além disso, eles mostram que o algoritmo proposto obtém, no pior caso, um limitante igual ao *roof dual*. Os resultados são apresentados para problemas com até 100 variáveis.

Uma abordagem trivial para obtenção de limitantes para o Problema Quadrático da Mochila (PQM) é apresentada em Billionnet e Calmels (1996). Essa abordagem é baseada na linearização do problema original e na resolução desse modelo pela relaxação linear de suas variáveis de decisão. Eles mostram que esses limitantes são de qualidade moderada. Além disso, eles utilizam a inserção de restrições de corte de *Chvatal-Gomory*, que apresentam melhores limitantes para o PQM.

Billionnet et al. (1999) propõem um limitante para o PQM baseado na divisão do conjunto de variáveis do problema original em diversos subconjuntos disjuntos. Essa divisão é feita aleatoriamente, gerando subconjuntos com no máximo 5 variáveis. A idéia é utilizar a Decomposição Lagrangiana para dividir o problema em subproblemas independentes. A escolha dos multiplicadores lagrangianos é feita por meio do Algoritmo de Subgradientes. Assim como apresentado por Chardaire e Sutter (1995), os autores mostram que quanto maior o tamanho dos subproblemas melhor é o limitante, entretanto, o tempo de processamento aumenta exponencialmente.

Billionnet e Elloumi (2007) aplicam técnicas de convexificação na função objetivo do PQ e aplicam o *software* CPLEX para resolver os modelos gerados. Os autores apresentam resultados para instâncias com até 200 variáveis. Pardalos e Jha (1992) discutem a complexidade computacional de vários problemas relacionados com o PQ e apresentam uma heurística de busca local para resolvê-los. Os resultados são apresentados para problemas com até 100 variáveis.

5.2 Formulação matemática

Seja $\mathbf{Q} = [q_{ij}]_{m \times m}$ uma matriz de números reais. O PQ pode ser formulado como apresentado na Equação 5.1.

$$PQ: \quad v(PQ) = \max_{x \text{ binário}} \sum_{i=1}^m \sum_{j=1}^m q_{ij} x_i x_j \quad (5.1)$$

Sem perder a generalidade do problema, a matriz \mathbf{Q} pode ser considerada simétrica (BILLIONNET; ELLOUMI, 2007). Logo, de forma análoga à apresentada por Adams e Dearing (1994), o PQ pode ser reescrito da seguinte forma:

$$PQ: \quad v(PQ) = \max_{x \text{ binário}} \left(\sum_{i=1}^m q_{ii} x_i + \sum_{(i,j) \in P \cup N} 2q_{ij} x_i x_j \right) \quad (5.2)$$

Onde $N = \{(i,j): i < j, q_{ij} < 0\}$ e $P = \{(i,j): i < j, q_{ij} > 0\}$.

Como apresentado em Adams e Dearing (1994), pode-se aplicar a técnica de linearização de Glover e Woolsey (1974) em 5.2, substituindo os termos quadráticos $x_i x_j$ pela variável contínua w_{ij} e por restrições que garantam que $w_{ij} = x_i x_j$. Logo, tem-se uma versão linear inteira mista de PQ (5.3 a 5.8). Por convenção, esse modelo será chamado PQL.

$PQL: \quad v(PQL) = \text{Maximizar:}$

$$\sum_{i=1}^m q_{ii} x_i + \sum_{(i,j) \in P \cup N} 2q_{ij} w_{ij} \quad (5.3)$$

Sujeito a:

$$w_{ij} - x_i \leq 0 \quad (i, j) \in P \quad (5.4)$$

$$w_{ij} - x_j \leq 0 \quad (i, j) \in P \quad (5.5)$$

$$x_i + x_j - w_{ij} \leq 1 \quad (i, j) \in N \quad (5.6)$$

$$w_{ij} \geq 0 \quad (i, j) \in N \quad (5.7)$$

$$x_i \in \{0, 1\} \quad i = 1, \dots, m \quad (5.8)$$

Fica claro nas funções 5.1, 5.2 e 5.3 que o número de elementos da matriz \mathbf{Q} utilizados em 5.2 e 5.3 será significativamente menor do que em 5.1, pois nesses casos são considerados apenas os elementos não nulos situados na metade superior e na diagonal principal dessa matriz. Um exemplo da utilização desses modelos é apresentado no Apêndice A.

5.3 Relaxações Lagrangianas tradicionais

A Relaxação Lagrangiana (ver Subseção 2.8.1) das restrições 5.4, 5.5 e/ou 5.6 pode ser utilizada para obtenção de limitantes para o PQ. Sendo assim, essas restrições podem ser combinadas de forma a gerar sete modelos diferentes para o problema:

- $L_\alpha PQL$: restrição 5.4 relaxada no sentido lagrangiano.
- $L_\beta PQL$: restrição 5.5 relaxada no sentido lagrangiano.
- $L_\lambda PQL$: restrição 5.6 relaxada no sentido lagrangiano.
- $L_{\alpha\beta} PQL$: restrições 5.4 e 5.5 relaxadas no sentido lagrangiano.
- $L_{\alpha\lambda} PQL$: restrições 5.4 e 5.6 relaxadas no sentido lagrangiano.
- $L_{\beta\lambda} PQL$: restrições 5.5 e 5.6 relaxadas no sentido lagrangiano.
- $L_{\alpha\beta\lambda} PQL$: restrições 5.4, 5.5 e 5.6 relaxadas no sentido lagrangiano.

Para cada um desses modelos, pode-se obter um limitante para o PQ por meio da minimização do seu dual lagrangiano (ver Subseção 2.8.1) correspondente ($DL_\alpha PQL$, $DL_\beta PQL$, etc). Um exemplo da utilização dessas relaxações é apresentado na Seção A.1.

5.4 Outras relaxações

O valor da solução da relaxação linear do PQL (substituir a restrição 5.8 por $0 \leq x_i \leq 1$) é um limitante trivial para o PQ (BILLIONNET; CALMELS, 1996). Como apresentado em Adams e Dearing (1994), esse limitante é conhecido como *roof dual*.

Um outro limitante para o PQ pode ser obtido pela inserção de uma restrição de corte de *Chvatal-Gomory* na relaxação linear de PQL. Essa restrição é descrita da seguinte maneira:

$$x_i + x_j + x_k - w_{ij} - w_{jk} - w_{ik} \leq 1 \quad (i, j), (j, k), (i, k) \in P \cup N \quad (5.9)$$

Essa restrição de corte é baseada em uma das restrições de corte apresentadas em Billionnet e Calmels (1996) para o Problema Quadrático da Mochila (PQM). Esses autores utilizam características baseadas em *cliques* (WOLSEY, 1998) e na generalização da restrição 5.6, porém considerando a ligação, por meio de arestas, entre três vértices no grafo. A Figura 5.1 apresenta uma situação na qual essa restrição é formada.

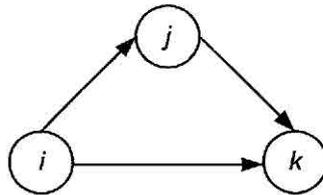


Figura 5.1 - Vértices e arestas necessárias para formação da restrição de corte.

A adaptação da restrição proposta em relação à apresentada em Billionnet e Calmels (1996) se deve principalmente ao fato de que o modelo PQL utiliza apenas parte da matriz de coeficientes \mathbf{Q} , e conseqüentemente o grafo gerado é direcionado. Dessa forma, como pode ser observado na Figura 5.1, e na definição do modelo PQL, não existe a aresta (k, i) , o que indica que $i < k$, e assim a aresta entre esses dois vértices será (i, k) .

Como mostrado por Billionnet e Calmels (1996), os limitantes obtidos com a inserção de restrições de corte é melhor do que o apresentado pela relaxação linear do PQM. Esse fato também é verificado para o PQL (ver Tabela 5.1). Por convenção, o modelo PQL com a restrição de corte será tratado adiante como PQLC. Um exemplo da utilização dessas relaxações é apresentado na Seção A.1.

5.5 Relaxações Lagrangianas com *Clusters*

A partir da matriz \mathbf{Q} descrita na seção anterior, pode-se criar um grafo $G=(V,A)$ com $V = \{1, \dots, m\}$ e $A = [a_{ij}]_{m \times m}$, $a_{ij} = 1$ se $q_{ij} \neq 0$ e $a_{ij} = 0$ se $q_{ij} = 0$. Particionando o grafo G em n ($n \leq m$) *clusters* independentes, com o uso das heurísticas disponíveis na biblioteca METIS (KARYPIS; KUMAR, 1998), tem-se $V = V_1 \cup V_2 \cup \dots \cup V_n$, onde $V_i \cap V_j = \emptyset$ ($\forall i, j = 1, \dots, n, i \neq j$), $G_i = (V_i, A_i)$, $i = 1, \dots, n$, e $X_i = V - V_i$, $i = 1, \dots, n$. A partir de então, o modelo PQL pode ser reescrito da seguinte forma:

PQL^n : $v(PQL^n) = \text{Maximizar:}$

$$\sum_{k=1}^n \left(\sum_{i \in V_k} q_{ii} x_i + \sum_{\substack{(i,j) \in P \cup N; \\ i \in V_k; j \in V_k \cup X_k}} 2q_{ij} w_{ij} \right) \quad (5.10)$$

Sujeito a:

$$w_{ij} - x_i \leq 0 \quad (i, j) \in P, i \in V_k, j \in V_k, k = 1, \dots, n \quad (5.11)$$

$$w_{ij} - x_j \leq 0 \quad (i, j) \in P, i \in V_k, j \in V_k, k = 1, \dots, n \quad (5.12)$$

$$x_i + x_j - w_{ij} \leq 1 \quad (i, j) \in N, i \in V_k, j \in V_k, k = 1, \dots, n \quad (5.13)$$

$$w_{ij} \geq 0 \quad (i, j) \in N, i \in V_k, j \in V_k, k = 1, \dots, n \quad (5.14)$$

$$w_{ij} - x_i \leq 0 \quad (i, j) \in P, i \in V_k, j \in X_k, k = 1, \dots, n \quad (5.15)$$

$$w_{ij} - x_j \leq 0 \quad (i, j) \in P, i \in V_k, j \in X_k, k = 1, \dots, n \quad (5.16)$$

$$x_i + x_j - w_{ij} \leq 1 \quad (i, j) \in N, i \in V_k, j \in X_k, k = 1, \dots, n \quad (5.17)$$

$$w_{ij} \geq 0 \quad (i, j) \in N, i \in V_k, j \in X_k, k = 1, \dots, n \quad (5.18)$$

$$x_i \in \{0, 1\} \quad i \in V_k, k = 1, \dots, n \quad (5.19)$$

Pode-se notar que as restrições 5.11 a 5.14 tratam apenas as arestas (i, j) cujos vértices são internos ao *cluster* (sub-grafo) k (arestas *intra-cluster*). Já as restrições 5.15 a 5.18 tratam as arestas (i, j) cujos vértices estão em *clusters* distintos (arestas *inter-clusters* ou arestas de ligação).

5.5.1 1ª abordagem

A primeira abordagem baseada na LagClus consiste em relaxar as restrições 5.15, 5.16 e 5.17 (arestas de ligação) no sentido lagrangiano por meio de vetores de multiplicadores $\vec{\alpha}$, $\vec{\beta}$ e $\vec{\lambda}$, respectivamente. Assim, tem-se três novos termos (um referente a cada restrição relaxada) que deverão ser inseridos na função objetivo 5.10. Esses termos são apresentados a seguir:

$$\sum_{k=1}^n \left(\sum_{\substack{(i,j) \in P; \\ i \in V_k; j \in X_k}} \alpha_{ij}(w_{ij} - x_i) \right), \quad \alpha_{ij} \geq 0 \quad (5.20)$$

$$\sum_{k=1}^n \left(\sum_{\substack{(i,j) \in P; \\ i \in V_k; j \in X_k}} \beta_{ij}(w_{ij} - x_j) \right), \quad \beta_{ij} \geq 0 \quad (5.21)$$

$$\sum_{k=1}^n \left(\sum_{\substack{(i,j) \in N; \\ i \in V_k; j \in X_k}} \lambda_{ij}(x_i + x_j - w_{ij} - 1) \right), \quad \lambda_{ij} \geq 0 \quad (5.22)$$

Após a retirada das restrições 5.15, 5.16 e 5.17, e a inserção desses termos na função objetivo (5.10), tem-se o problema PQL^n (indiretamente o PQ) dividido em n subproblemas independentes, sendo cada subproblema k ($k = 1, \dots, n$) representado pelo modelo descrito a seguir.

$LC_{\alpha\beta\lambda}PQL_k$: $v(LC_{\alpha\beta\lambda}PQL_k) = \text{Maximizar:}$

$$\begin{aligned} & \sum_{i \in V_k} q_{ii}x_i + \sum_{(i,j) \in (P \cup N) \cap A_k} 2q_{ij}w_{ij} + \sum_{\substack{(i,j) \in P \\ i \in V_k, j \in X_k}} \alpha_{ij}x_i + \sum_{\substack{(i,j) \in P \\ i \in X_k, j \in V_k}} \beta_{ij}x_j \\ & - \sum_{\substack{(i,j) \in N \\ i \in V_k, j \in X_k}} \lambda_{ij}x_i - \sum_{\substack{(i,j) \in N \\ i \in X_k, j \in V_k}} \lambda_{ij}x_j + \sum_{\substack{(i,j) \in P \\ i \in V_k, j \in X_k}} (2q_{ij} - \alpha_{ij} - \beta_{ij})w_{ij} \\ & + \sum_{\substack{(i,j) \in N \\ i \in V_k, j \in X_k}} (2q_{ij} + \lambda_{ij})w_{ij} + \sum_{\substack{(i,j) \in N; \\ i \in V_k, j \in X_k}} \lambda_{ij} \end{aligned} \quad (5.23)$$

Sujeito a:

$$w_{ij} - x_i \leq 0 \quad (i, j) \in P, i \in V_k, j \in V_k \quad (5.24)$$

$$w_{ij} - x_j \leq 0 \quad (i, j) \in P, i \in V_k, j \in V_k \quad (5.25)$$

$$x_i + x_j - w_{ij} \leq 1 \quad (i, j) \in N, i \in V_k, j \in V_k \quad (5.26)$$

$$w_{ij} \geq 0 \quad (i, j) \in N, i \in V_k, j \in V_k \quad (5.27)$$

$$w_{ij} \in \{0, 1\} \quad (i, j) \in P \cup N, i \in V_k, j \in X_k \quad (5.28)$$

$$x_i \in \{0, 1\} \quad i \in V_k \quad (5.29)$$

A restrição 5.18 é substituída pela restrição 5.28 para manter a viabilidade das soluções dos subproblemas. Por fim, a solução da relaxação do problema PQL (que é equivalente a PQ) em n subproblemas (*clusters*) é dada pela Equação 5.30, e o seu dual lagrangiano é apresentado na Equação 5.31.

$$LC_{\alpha\beta\lambda}PQL^n : v(LC_{\alpha\beta\lambda}PQL^n) = \sum_{k=1}^n v(LC_{\alpha\beta\lambda}PQL_k) + \sum_{\substack{(i,j) \in N; \\ d(i) \neq d(j)}} \lambda_{ij} \quad (5.30)$$

$$DLC_{\alpha\beta\lambda}PQL^n : v(DLC_{\alpha\beta\lambda}PQL^n) = \text{Min}\{v(LC_{\alpha\beta\lambda}PQL^n)\}, \alpha, \beta, \lambda \geq 0. \quad (5.31)$$

Como demonstrado por Adams e Dearing (1994), para $n = m$ tem-se que $DLC_{\alpha\beta\lambda}PQL^n$ é o *roof dual* para o PQ, e conseqüentemente, pode-se afirmar que:

$$v(DLC_{\alpha\beta\lambda}PQL^n) \leq v(DLC_{\alpha\beta\lambda}PQL^m) = \text{roof dual} \quad (5.32)$$

5.5.2 2ª abordagem

Seguindo a 1ª abordagem, surge a idéia de não relaxar a restrição 5.15, pois assim como na abordagem anterior, as variáveis w_{ij} e x_i serão tratadas no subproblema onde se encontra o vértice i . Logo, a relaxação dessa restrição não necessariamente deve ser considerada. Sendo assim, uma outra forma de relaxação do PQL em *clusters* pode ser escrita como:

$$LC_{\beta\lambda}PQL_k : v(LC_{\beta\lambda}PQL_k) = \text{Maximizar:}$$

$$\begin{aligned} & \sum_{i \in V_k} q_{ii}x_i + \sum_{\substack{(i,j) \in P \cup N; \\ i \in V_k; j \in V_k \cup X_k}} 2q_{ij}w_{ij} + \sum_{\substack{(i,j) \in P; \\ j \in V_k; i \in X_k}} \beta_{ij}x_j - \sum_{\substack{(i,j) \in N; \\ i \in V_k; j \in X_k}} \lambda_{ij}x_i \\ & - \sum_{\substack{(i,j) \in N; \\ i \in X_k; j \in V_k}} \lambda_{ij}x_j - \sum_{\substack{(i,j) \in P; \\ i \in V_k; j \in X_k}} \beta_{ij}w_{ij} + \sum_{\substack{(i,j) \in P; \\ i \in V_k; j \in X_k}} \lambda_{ij}w_{ij} \end{aligned} \quad (5.33)$$

Sujeito a:

$$w_{ij} - x_i \leq 0 \quad (i, j) \in P, i \in V_k, j \in V_k \cup X_k \quad (5.34)$$

$$w_{ij} - x_j \leq 0 \quad (i, j) \in P, i \in V_k, j \in V_k \quad (5.35)$$

$$x_i + x_j - w_{ij} \leq 1 \quad (i, j) \in N, i \in V_k, j \in V_k \quad (5.36)$$

$$w_{ij} \geq 0 \quad (i, j) \in N, i \in V_k, j \in V_k \quad (5.37)$$

$$w_{ij} \in \{0, 1\} \quad (i, j) \in N, i \in V_k, j \in X_k \quad (5.38)$$

$$x_i \in \{0, 1\} \quad i \in V_k \quad (5.39)$$

A restrição 5.34 forma uma “junção” das restrições 5.11 e 5.15. Logo, a solução da relaxação do problema PQL em n clusters é dada pela Equação 5.40, e o seu dual lagrangiano pela Equação 5.41.

$$LC_{\beta\lambda}PQL^n : v(LC_{\beta\lambda}PQL^n) = \sum_{k=1}^n v(LC_{\beta\lambda}PQL_k) + \sum_{\substack{(i,j) \in N; \\ d(i) \neq d(j)}} \lambda_{ij} \quad (5.40)$$

$$DLC_{\beta\lambda}PQL^n : v(DLC_{\beta\lambda}PQL^n) = \mathbf{Min}\{v(LC_{\beta\lambda}PQL^n)\}, \beta, \lambda \geq 0. \quad (5.41)$$

Mais uma vez, a partir das idéias expostas em Adams e Dearing (1994), e da 1ª abordagem, pode-se afirmar diretamente que:

$$\begin{aligned} v(DLC_{\beta\lambda}PQL^n) &\leq v(DLC_{\alpha\beta\lambda}PQL^n) \leq \\ v(DLC_{\beta\lambda}PQL^m) &= v(DLC_{\alpha\beta\lambda}PQL^m) = \text{roof dual} \end{aligned} \quad (5.42)$$

5.5.3 Análise das abordagens propostas

As duas abordagens baseadas na LagClus são muito semelhantes, porém a 2ª apresenta uma relaxação mais forte em relação a 1ª devido a não relaxação da restrição 5.15. Um exemplo de aplicação dessas abordagens é apresentado na Seção A.2.

5.6 Decomposições Lagrangianas

Na busca por relaxações ainda mais fortes para o PQ, são propostos diferentes modelos de relaxações baseados na técnica de Decomposição Lagrangiana (ver Subseção 2.8.3).

5.6.1 1ª abordagem

A partir do grafo G (ver Seção 5.5), o PQL é decomposto em n subproblemas por meio da cópia das arestas formadas por vértices pertencentes a *clusters* distintos (arestas *inter-clusters*) e de seus respectivos vértices. Além disso, o valor associado (q_{ij}) a cada aresta é dividido entre os originais e suas cópias. Logo, o PQL pode ser reescrito da seguinte forma:

PQL^n : $v(PQL^n) = \text{Maximizar:}$

$$\sum_{k=1}^n \left(\sum_{i \in V_k} \frac{q_{ii}}{c_i} x_i + \sum_{\substack{\exists(i,j), \\ i \in V_k, j \in X_k}} \frac{q_{ij}}{c_j} x_{j'}^k + \sum_{(i,j) \in (P \cup N) \cap A_k} 2q_{ij} w_{ij} \right) \quad (5.43)$$

$$+ \sum_{k=1}^n \left(\sum_{\substack{(i,j) \in P \cup N; \\ i \in V_k, j \in X_k}} \frac{2q_{ij}}{2} w_{ij} + \sum_{\substack{(i,j) \in P \cup N; \\ i \in V_k, j \in X_k}} \frac{2q_{ii}}{2} w_{ij'} \right)$$

Sujeito a:

$$w_{ij} - x_i \leq 0 \quad (i, j) \in P, i \in V_k, j \in V_k, k = 1, \dots, n \quad (5.44)$$

$$w_{ij} - x_j \leq 0 \quad (i, j) \in P, i \in V_k, j \in V_k, k = 1, \dots, n \quad (5.45)$$

$$x_i + x_j - w_{ij} \leq 1 \quad (i, j) \in N, i \in V_k, j \in V_k, k = 1, \dots, n \quad (5.46)$$

$$w_{ij} \geq 0 \quad (i, j) \in N, i \in V_k, j \in V_k, k = 1, \dots, n \quad (5.47)$$

$$w_{ij'} - x_i \leq 0 \quad (i, j) \in P, i \in V_k, j \in X_k, k = 1, \dots, n \quad (5.48)$$

$$w_{ij'} - x_{j'}^k \leq 0 \quad (i, j) \in P, i \in V_k, j \in X_k, k = 1, \dots, n \quad (5.49)$$

$$x_i + x_{j'}^k - w_{ij'} \leq 1 \quad (i, j) \in N, i \in V_k, j \in X_k, k = 1, \dots, n \quad (5.50)$$

$$w_{ij'} \geq 0 \quad (i, j) \in N, i \in V_k, j \in X_k, k = 1, \dots, n \quad (5.51)$$

$$x_j = x_{j'}^k \quad j \in X_k, k = 1, \dots, n \quad (5.52)$$

$$w_{ij} = w_{ij'} \quad (i, j) \in P \cup N, i \in V_k, j \in X_k, k = 1, \dots, n \quad (5.53)$$

$$x_i \in \{0, 1\} \quad i \in V_k, k = 1, \dots, n \quad (5.54)$$

Nesse modelo, as variáveis $x_{j'}^k$ representam as cópias do vértice j (j') no *cluster* k , e

as variáveis w_{ij} representam as arestas entre os vértices i e j' (cópia de j). c_i indica a quantidade de cópias do vértice i existente nos n clusters.

As restrições 5.44 a 5.47 tratam apenas as arestas (i,j) cujos vértices são internos ao cluster (sub-grafo) k (arestas *intra-cluster*). Já as restrições 5.48 a 5.51 tratam as arestas (i,j) cujos vértices estão em clusters distintos (arestas *inter-clusters* - arestas de ligação). As restrições 5.52 e 5.53 são as restrições de cópia, que garantem a igualdade entre as variáveis originais e suas cópias.

Relaxando as restrições 5.52 e 5.53 no sentido lagrangiano por meio de vetores de multiplicadores $\vec{\alpha}$ e $\vec{\beta}$, respectivamente, tem-se dois novos termos (um referente a cada restrição relaxada) que deverão ser inseridos na função objetivo 5.43. Esses termos são apresentados a seguir:

$$\sum_{k=1}^n \left(\sum_{\substack{\exists(i,j), \\ i \in V_k; j \in X_k}} \alpha_j^k (x_j - x_{j'}) \right), \quad \alpha_j^k \text{ irrestrito} \quad (5.55)$$

$$\sum_{k=1}^n \left(\sum_{\substack{(i,j) \in P \cup N; \\ i \in V_k; j \in X_k}} \beta_{ij} (w_{ij} - w_{ij'}) \right), \quad \beta_{ij} \text{ irrestrito} \quad (5.56)$$

Assim, após a retirada das restrições de cópia, e a inserção desses termos na função objetivo, tem-se o problema PQL^n (indiretamente o PQ) dividido em n subproblemas independentes, sendo cada subproblema k ($k = 1, \dots, n$) representado pelo modelo descrito a seguir.

$DC_{\alpha\beta}PQL_k$: $v(DC_{\alpha\beta}PQL_k) = \text{Maximizar:}$

$$\begin{aligned} & \sum_{i \in V_k} \left(\frac{q_{ii}}{c_i} - \sum_{d \neq k} \alpha_i^d \right) x_i + \sum_{\substack{\exists(i,j), \\ i \in V_k; j \in X_k}} \left(\frac{q_{jj}}{c_j} + \alpha_j^k \right) x_{j'} + \sum_{\substack{(i,j) \in \\ (P \cup N) \cap A_k}} 2q_{ij} w_{ij} \\ & + \sum_{\substack{(i,j) \in P \cup N; \\ i \in V_k; j \in X_k}} (q_{ij} - \beta_{ij}) w_{ij} + \sum_{\substack{(i,j) \in P \cup N; \\ i \in V_k; j \in X_k}} (q_{ij} + \beta_{ij}) w_{ij'} \end{aligned} \quad (5.57)$$

Sujeito a:

$$w_{ij} - x_i \leq 0 \quad (i, j) \in P, i \in V_k, j \in V_k \quad (5.58)$$

$$w_{ij} - x_j \leq 0 \quad (i, j) \in P, i \in V_k, j \in V_k \quad (5.59)$$

$$x_i + x_j - w_{ij} \leq 1 \quad (i, j) \in N, i \in V_k, j \in V_k \quad (5.60)$$

$$w_{ij} \geq 0 \quad (i, j) \in N, i \in V_k, j \in V_k \quad (5.61)$$

$$w_{ij'} - x_i \leq 0 \quad (i, j) \in P, i \in V_k, j \in X_k \quad (5.62)$$

$$w_{ij'} - x_{j'}^k \leq 0 \quad (i, j) \in P, i \in V_k, j \in X_k \quad (5.63)$$

$$x_i + x_{j'}^k - w_{ij'} \leq 1 \quad (i, j) \in N, i \in V_k, j \in X_k \quad (5.64)$$

$$w_{ij} \in \{0, 1\} \quad (i, j) \in N, i \in V_k, j \in X_k \quad (5.65)$$

$$w_{ij'} \in \{0, 1\} \quad (i, j) \in N, i \in V_k, j \in X_k \quad (5.66)$$

$$x_{j'}^k \in \{0, 1\} \quad j \in X_k \quad (5.67)$$

$$x_i \in \{0, 1\} \quad i \in V_k \quad (5.68)$$

As restrições 5.65, 5.66 e 5.67 são inseridas no modelo para manter a viabilidade das soluções dos subproblemas. Por fim, a solução da relaxação do problema PQL (que é equivalente a PQ) em n subproblemas (*clusters*), é dada pela Equação 5.69, e o seu dual lagrangiano é apresentado na Equação 5.70.

$$DC_{\alpha\beta}PQL^n : \quad v(DC_{\alpha\beta}PQL^n) = \sum_{k=1}^n v(DC_{\alpha\beta}PQL_k) \quad (5.69)$$

$$DDC_{\alpha\beta}PQL^n : v(DDC_{\alpha\beta}PQL^n) = \mathbf{Min}\{v(DC_{\alpha\beta}PQL^n)\} \quad (5.70)$$

α, β irrestritos

5.6.2 2ª abordagem

Seguindo a 1ª abordagem (ver Subseção 5.6.1), surge a idéia de substituir as variáveis w_{ij} ($i \in V_k$ e $j \in X_k$) pelas variáveis $w_{ij'}$, e não dividir o valor associado às arestas. Logo, o PQL pode ser escrito da seguinte forma:

PQL^n : $v(PQL^n) = \text{Maximizar:}$

$$\sum_{k=1}^n \left(\sum_{i \in V_k} q_{ii} x_i + \sum_{(i,j) \in (P \cup N) \cap A_k} 2q_{ij} w_{ij} + \sum_{\substack{(i,j) \in P \cup N; \\ i \in V_k; j \in X_k}} 2q_{ij} w_{ij'} \right) \quad (5.71)$$

Sujeito a: (5.44),(5.45),...,5.52 e 5.54;

Nesse modelo, não existe a restrição de cópia (5.53) das variáveis w_{ij} ($i \in V_k$ e $j \in X_k$), pois essas são substituídas diretamente pelas variáveis $w_{ij'}$. Relaxando a restrição de cópia (5.52) no sentido lagrangiano por meio do vetor de multiplicadores $\vec{\alpha}$, o problema também pode ser dividido em n subproblemas independentes, sendo:

$DC_\alpha PQL_k$: $v(DC_\alpha PQL_k) = \text{Maximizar:}$

$$\sum_{i \in V_k} \left(q_{ii} - \sum_{d \neq k} \alpha_i^d \right) x_i + \sum_{\substack{\exists (i,j), i \in V_k; \\ j \in X_k}} \alpha_j^k x_j^k + \sum_{\substack{(i,j) \in \\ (P \cup N) \cap A_k}} 2q_{ij} w_{ij} + \sum_{\substack{(i,j) \in P \cup N; \\ i \in V_k; j \in X_k}} 2q_{ij} w_{ij'} \quad (5.72)$$

Sujeito a: (5.58),(5.59),..., (5.64),(5.67) e (5.68)

$$w_{ij'} \geq 0 \quad (i, j) \in N, i \in V_k, j \in X_k \quad (5.73)$$

A restrição 5.73 é inserida no modelo, e nesse caso ela não precisa ser binária para manter a viabilidade das soluções dos subproblemas. A solução da relaxação do problema em n subproblemas (*clusters*), é dada pela Equação 5.74, e o seu dual lagrangiano pela Equação 5.75.

$$DC_\alpha PQL^n : \quad v(DC_\alpha PQL^n) = \sum_{k=1}^n v(DC_\alpha PQL_k) \quad (5.74)$$

$$DDC_\alpha PQL^n : \quad v(DDC_\alpha PQL^n) = \text{Min}\{v(DC_\alpha PQL^n)\}, \alpha \text{ irrestrito} \quad (5.75)$$

5.6.3 3ª abordagem

Seguindo a 2ª abordagem, e as idéias apresentadas em Billionnet et al. (1999) e Chardaire e Sutter (1995), surge a idéia de utilizar todos os elementos da matriz \mathbf{Q} (ver Seção 5.2) no modelo linear do PQ (PQL). Por um questão de notação, esse modelo será tratado como PQLT.

PQLT: $v(PQLT) = \text{Maximizar:}$

$$\sum_{i=1}^m q_{ii}x_i + \sum_{i \neq j} q_{ij}w_{ij} \quad (5.76)$$

Sujeito a:

$$w_{ij} - x_i \leq 0 \quad q_{ij} > 0 \quad (5.77)$$

$$w_{ij} - x_j \leq 0 \quad q_{ij} > 0 \quad (5.78)$$

$$x_i + x_j - w_{ij} \leq 1 \quad q_{ij} < 0 \quad (5.79)$$

$$w_{ij} \geq 0 \quad q_{ij} < 0 \quad (5.80)$$

$$x_i \in \{0, 1\} \quad i = 1, \dots, m \quad (5.81)$$

A partir dessa formulação, pode-se também obter um grafo G , semelhante ao descrito na Seção 5.5, porém agora com duas arestas entre os vértices, ou seja, (i, j) e (j, i) . Assim, o PQLT é decomposto em n subproblemas por meio da cópia das arestas formadas por vértices pertencentes a *clusters* distintos (arestas *inter-clusters*) e de seus respectivos vértices. Logo, o PQLT pode ser reescrito da seguinte forma:

PQLTⁿ: $v(PQLT^n) = \text{Maximizar:}$

$$\sum_{k=1}^n \left(\sum_{i \in V_k} q_{ii}x_i + \sum_{\substack{i \in V_k, j \in V_k, \\ i \neq j}} q_{ij}w_{ij} + \sum_{\substack{i \in V_k, j \in X_k, \\ i \neq j}} q_{ij}w_{ij}' \right) \quad (5.82)$$

Sujeito a:

$$w_{ij} - x_i \leq 0 \quad i \in V_k, j \in V_k, i \neq j, q_{ij} > 0, k = 1, \dots, n \quad (5.83)$$

$$w_{ij} - x_j \leq 0 \quad i \in V_k, j \in V_k, i \neq j, q_{ij} > 0, k = 1, \dots, n \quad (5.84)$$

$$x_i + x_j - w_{ij} \leq 1 \quad i \in V_k, j \in V_k, i \neq j, q_{ij} < 0, k = 1, \dots, n \quad (5.85)$$

$$w_{ij} \geq 0 \quad i \in V_k, j \in V_k, i \neq j, q_{ij} < 0, k = 1, \dots, n \quad (5.86)$$

$$w_{ij}' - x_i \leq 0 \quad i \in V_k, j \in X_k, i \neq j, q_{ij} > 0, k = 1, \dots, n \quad (5.87)$$

$$w_{ij}' - x_j' \leq 0 \quad i \in V_k, j \in X_k, i \neq j, q_{ij} > 0, k = 1, \dots, n \quad (5.88)$$

$$x_i + x_{j'}^k - w_{ij'} \leq 1 \quad i \in V_k, j \in X_k, i \neq j, q_{ij} < 0, k = 1, \dots, n \quad (5.89)$$

$$w_{ij'} \geq 0 \quad i \in V_k, j \in X_k, i \neq j, q_{ij} < 0, k = 1, \dots, n \quad (5.90)$$

$$x_j = x_{j'}^k \quad j \in X_k, k = 1, \dots, n \quad (5.91)$$

$$w_{ij'} = w_{ji'} \quad i \in V_k, j \in X_k, j > i, k = 1, \dots, n \quad (5.92)$$

$$x_i \in \{0, 1\} \quad i \in V_k, k = 1, \dots, n \quad (5.93)$$

Relaxando as restrições 5.91 e 5.92 no sentido lagrangiano, por meio dos vetores de multiplicadores $\vec{\alpha}$ e $\vec{\beta}$, o problema pode ser dividido em n subproblemas independentes, sendo:

$DC_{\alpha\beta}PQLT_k$: $v(DC_{\alpha\beta}PQLT_k) = \text{Maximizar:}$

$$\sum_{i \in V_k} \left(q_{ii} - \sum_{d \neq k} \alpha_i^d \right) x_i + \sum_{j \in X_k} \alpha_j^k x_{j'}^k + \sum_{i \in V_k} \sum_{j \in V_k; j \neq i} q_{ij} w_{ij} \quad (5.94)$$

$$+ \sum_{i \in V_k} \sum_{\substack{j \in X_k; \\ j > i}} (q_{ij} - \beta_{ij}) w_{ij'} + \sum_{i \in V_k} \sum_{\substack{j \in X_k; \\ j < i}} (q_{ij} + \beta_{ji}) w_{ij'}$$

Sujeito a:

$$w_{ij} - x_i \leq 0 \quad i \in V_k, j \in V_k, i \neq j, q_{ij} > 0, k = 1, \dots, n \quad (5.95)$$

$$w_{ij} - x_j \leq 0 \quad i \in V_k, j \in V_k, i \neq j, q_{ij} > 0, k = 1, \dots, n \quad (5.96)$$

$$x_i + x_j - w_{ij} \leq 1 \quad i \in V_k, j \in V_k, i \neq j, q_{ij} < 0, k = 1, \dots, n \quad (5.97)$$

$$w_{ij} \geq 0 \quad i \in V_k, j \in V_k, i \neq j, q_{ij} < 0, k = 1, \dots, n \quad (5.98)$$

$$w_{ij'} - x_i \leq 0 \quad i \in V_k, j \in X_k, i \neq j, q_{ij} > 0, k = 1, \dots, n \quad (5.99)$$

$$w_{ij'} - x_{j'}^k \leq 0 \quad i \in V_k, j \in X_k, i \neq j, q_{ij} > 0, k = 1, \dots, n \quad (5.100)$$

$$x_i + x_{j'}^k - w_{ij'} \leq 1 \quad i \in V_k, j \in X_k, i \neq j, q_{ij} < 0, k = 1, \dots, n \quad (5.101)$$

$$w_{ij'} \in \{0, 1\} \quad i \in V_k, j \in X_k, i \neq j, q_{ij} < 0, k = 1, \dots, n \quad (5.102)$$

$$x_{j'}^k \in \{0, 1\} \quad j \in X_k \quad (5.103)$$

$$x_i \in \{0, 1\} \quad i \in V_k, k = 1, \dots, n \quad (5.104)$$

As restrições 5.102 e 5.103 são inseridas no modelo para manter a viabilidade das soluções dos subproblemas. Por fim, a solução da relaxação do problema PQLT em n subproblemas, é dada pela Equação 5.105, e o seu dual lagrangiano é apresentado na Equação 5.106.

$$DC_{\alpha\beta}PQLT^n : \quad v(DC_{\alpha\beta}PQLT^n) = \sum_{k=1}^n v(DC_{\alpha\beta}PQLT_k) \quad (5.105)$$

$$DDC_{\alpha\beta}PQLT^n : v(DDC_{\alpha\beta}PQLT^n) = \underset{\alpha, \beta \text{ irrestritos}}{\text{Min}} \{v(DC_{\alpha\beta}PQLT^n)\} \quad (5.106)$$

5.6.4 Análise das abordagens propostas

As três abordagens apresentadas nesta seção possuem idéias semelhantes, porém cada uma apresenta características próprias.

A 1ª abordagem copia as arestas formadas por vértices pertencentes a *clusters* distintos e seus respectivos vértices. Além disso, o valor associado a cada aresta é dividido entre as originais e suas cópias.

A 2ª abordagem substitui as variáveis w_{ij} ($i \in V_k$ e $j \in X_k$) pelas variáveis $w_{ij'}$, ou seja, as arestas (i, j) pelas arestas (i, j') e não divide o valor associado às arestas.

Já na 3ª abordagem, toda a matriz \mathbf{Q} é utilizada, e ambos os vértices referentes às arestas relaxadas são copiados para o *cluster* oposto. Nesse caso, os valores associados as arestas são mantidos no *cluster* do vértice de origem das respectivas arestas.

Um exemplo de aplicação dessas três abordagens é apresentado na Seção A.3.

5.7 Geração de Colunas

Seguindo os preceitos descritos na Seção 2.4, a técnica de Geração de Colunas é aplicada à LagClus e à DecLag. Para cada caso, foi utilizada a abordagem (modelo) que apresentou os melhores resultados (ver Seção 5.9).

5.7.1 GC aplicada à LagClus

A Geração de Colunas (GC) é aplicada à 2ª abordagem da LagClus (ver Subseção 5.5.2), pois como pode ser observado na Tabela 5.3, essa abordagem apresentou

melhores resultados em relação à 1ª. Para facilitar a compreensão dessa abordagem, o modelo apresentado para a LagClus (5.10 a 5.19) pode ser representado matricialmente da seguinte maneira:

PQL^n : $v(PQL^n) = \text{Maximizar:}$

$$\sum_{k=1}^n (\mathbf{q}_x^k \mathbf{x}^k + \mathbf{q}_w^k \mathbf{w}^k) \quad (5.107)$$

$$\text{Sujeito a : } \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_n \\ \mathbf{D}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}_n \end{bmatrix} \times \begin{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{w} \end{bmatrix}^1 \\ \begin{bmatrix} \mathbf{x} \\ \mathbf{w} \end{bmatrix}^2 \\ \vdots \\ \begin{bmatrix} \mathbf{x} \\ \mathbf{w} \end{bmatrix}^n \end{bmatrix} \approx \mathbf{b} \quad (5.108)$$

\mathbf{q}_x^k é um vetor com os coeficientes das variáveis x_i presentes no *cluster* k ; \mathbf{q}_w^k é um vetor com os coeficientes das variáveis w_{ij} presentes no *cluster* k ; D_k é uma matriz com os coeficientes das variáveis (pertencentes ao *cluster* k) presentes nas restrições *intra-cluster* (5.11 a 5.15); A_k é uma matriz com os coeficientes das variáveis (pertencentes ao *cluster* k) presentes restrições 5.16 e 5.17; \mathbf{x} é um vetor com os valores das variáveis de decisão x_i ; \mathbf{w} é um vetor com os valores das variáveis de decisão w_{ij} ; \approx são os operadores relacionais (\leq ou \geq), variando de acordo com a restrição; \mathbf{b} são os valores do lado direito das restrições (0 ou 1).

Relaxando as restrições presentes nas matrizes \mathbf{A} (5.16 e 5.17) no sentido lagrangiano, por meio do vetor de multiplicadores $\vec{\alpha}$, ($\vec{\alpha} \geq 0$), o problema PQL^n (indiretamente o PQ) pode ser dividido em n subproblemas independentes. Considerando R_k como sendo as restrições associadas ao subproblema k , $k=1, \dots, n$, e d o número de restrições relaxadas, cada subproblema pode ser representado como descrito em 5.109, e a solução da relaxação do problema PQL em n subproblemas (*clusters*) é dada pela Equação 5.110.

$$LC_\alpha PQL_k : \quad v(LC_\alpha PQL_k) = \max \{ (\mathbf{q}_x^k - \vec{\alpha} \mathbf{A}_k) \mathbf{x}^k + (\mathbf{q}_w^k - \vec{\alpha} \mathbf{A}_k) \mathbf{w}^k : \mathbf{x}^k, \mathbf{w}^k \in R_k \} \quad (5.109)$$

$$LC_\alpha PQL^n : v(LC_\alpha PQL^n) = \sum_{k=1}^n v(LC_\alpha PQL_k) - \sum_{p=1}^d \alpha_p \quad (5.110)$$

Logo, de acordo com as idéias da Decomposição Dantzig-Wolfe (ver Seção 2.3), pode-se gerar um Problema Mestre Restrito (PMR) assim como descrito a seguir.

$$PMR_{PL}^{LC} : v(PMR_{PL}^{LC}) = \text{Maximizar:}$$

$$\sum_{k=1}^n \lambda_{k,t} (\mathbf{q}_x^{k,t} \mathbf{x}^{k,t} + \mathbf{q}_w^{k,t} \mathbf{w}^{k,t}) \quad (5.111)$$

Sujeito a:

$$\sum_{k=1}^n \sum_{t \in T_k} \lambda_{k,t} (\mathbf{A}_k \mathbf{y}^{k,t}) \approx \mathbf{b}, \text{ onde } \mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \mathbf{w} \end{bmatrix} \quad (5.112)$$

$$\sum_{t \in T_k} \lambda_{k,t} = 1 \quad \forall k \in \{1, \dots, n\} \quad (5.113)$$

$$\lambda_{k,t} \geq 0 \quad \forall k \in \{1, \dots, n\}, t \in T_k \quad (5.114)$$

O conjunto de pontos extremos associados com as colunas geradas no PMR é dado por T_k . μ_k é a variável dual associada com a k -ésima restrição de convexidade de 5.113. $\mathbf{x}^{k,t}$ e $\mathbf{w}^{k,t}$ são vetores que definem os pontos extremos $t \in T_k$, ou seja, as soluções viáveis do subproblema definido pelo *cluster* k . $\lambda_{k,t}$ é a variável de decisão correspondente ao ponto extremo $t \in T_k$ referente ao subproblema k . Para cada subproblema k (5.109), pode-se substituir o vetor de multiplicadores lagrangianos $\bar{\alpha}$ pelo vetor de variáveis duais $\bar{\pi}$ associado com as restrições 5.112, e de uma forma alternativa, cada subproblema pode ser descrito pela Equação 5.115, e a relaxação do PQL, em n *clusters*, pela Equação 5.116.

$$Z_k^{LC} = \max \{ (\mathbf{q}_x^{k,t} - \bar{\pi} \mathbf{A}_k) \mathbf{x}^{k,t} + (\mathbf{q}_w^{k,t} - \bar{\pi} \mathbf{A}_k) \mathbf{w}^{k,t} : \mathbf{x}^{k,t}, \mathbf{w}^{k,t} \in R_k \} \quad (5.115)$$

$$LC_\pi PQL^n : v(LC_\pi PQL^n) = \sum_{k=1}^n Z_k^{LC} + \sum_{p=1}^d \pi_p \quad (5.116)$$

A partir de então, uma nova coluna gerada pelo subproblema k é inserida no PMR se o seu custo reduzido θ_k for positivo, isto é, $\theta_k = Z_k^{LC} - \mu_k > 0$. Assim, o PMR coordena as soluções dos subproblemas por meio de suas variáveis duais, buscando uma solução para o problema original.

O PMR inicial é gerado pela heurística apresentada na Figura 5.2. Essa heurística utiliza a busca local apresentada em Beasley (1998). Em seguida, novas colunas são geradas até que um critério de parada seja satisfeito, e o PMR final, formado por todas as colunas geradas, é resolvido de forma inteira ($\lambda_{k,t} \in \{0, 1\}$), e consequentemente, sua solução será equivalente a uma solução viável para o PQL, e indiretamente para o PQ. Por convenção, esse problema será tratado como PMR_{PLI}^{LC} , e o valor de sua solução como $v(PMR_{PLI}^{LC})$. A Figura 5.3 apresenta os procedimentos para execução da GC.

```

1. PARA v=1 ATÉ m FAÇA
2.    $x_v \leftarrow 1$ ;
3.   APLICAR (busca local);
4.   PARA (cada variável  $w_{ij}$ ) FAÇA
5.     SE ( $x_i = 1$  E  $x_j = 1$ ) ENTÃO
6.        $w_{ij} \leftarrow 1$ ;
7.     SENÃO
8.        $w_{ij} \leftarrow 0$ ;
9.     FIM-SE;
10.  FIM-PARA;
11.  DIVIDIR (as variáveis da solução nos respectivos clusters);
12.  CRIAR (uma coluna para cada cluster);
13.  INSERIR (as colunas no PMR);
14. FIM-PARA;

```

Figura 5.2 - Formação do PMR inicial na Geração de Colunas com a LagClus.

A solução do PMR de forma inteira - $v(PMR_{PLI}^{LC})$ - apresenta uma solução viável para o PQ. Já a solução da relaxação linear do PMR - $v(PMR_{PL}^{LC})$ - e a solução da relaxação com divisão em *clusters* - $v(LC_{\pi}PQL^n)$ - apresentam limitantes para o PQ. Um exemplo de aplicação dessa abordagem é apresentado na Subseção A.4.1.

```

1. FORMAR (o PMR inicial);
2. OTIMIZAR (o PMR via CPLEX e obter as variáveis duais);
3. ATUALIZAR (a função objetivo dos subproblemas);
4. RESOLVER (os subproblemas via CPLEX);
5. SE ( $\forall k, \theta_k \leq 0$ ) OU  $v(LC_{\pi}PQL^n) = v(PMR_{PL}^{LC})$  ENTÃO
6.   PARAR, pois a solução ótima para o PMR foi encontrada;
7.   RESOLVER (o PMR de forma inteira via CPLEX - obter  $v(PMR_{PLI}^{LC})$ );
8. SENÃO
9.   INSERIR (toda coluna cujo  $\theta_k > 0$  no PMR);
10.  VOLTAR (ao passo 2);
11. FIM-SE;

```

Figura 5.3 - Passos para execução da Geração de Colunas com a LagClus.

5.7.2 GC aplicada à DecLag

Seguindo a idéia apresentada na seção anterior, a Geração de Colunas (GC) também é aplicada à 3ª abordagem de Decomposição Lagrangiana proposta (ver Subseção 5.6.3). Mais uma vez, a escolha dessa abordagem para aplicação da GC se deve aos resultados obtidos nos experimentos computacionais realizados (ver Tabela 5.4). Assim, de forma análoga à apresentada na seção anterior, o modelo (5.82 a 5.93) pode ser representado matricialmente da seguinte maneira:

$PQLT^n$: $v(PQLT^n) = \text{Maximizar:}$

$$\sum_{k=1}^n (\mathbf{q}_x^k \mathbf{x}^k + \mathbf{0} \mathbf{x}'^k + \mathbf{q}_w^k \mathbf{w}^k + \mathbf{q}_{w'}^k \mathbf{w}'^k) \quad (5.117)$$

$$\text{Sujeito a : } \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_n \\ \mathbf{D}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}_n \end{bmatrix} \times \begin{bmatrix} \left[\begin{array}{c} \mathbf{x} \\ \mathbf{x}' \\ \mathbf{w} \\ \mathbf{w}' \\ \vdots \\ \mathbf{x} \\ \mathbf{x}' \\ \mathbf{w} \\ \mathbf{w}' \end{array} \right]^1 \\ \vdots \\ \left[\begin{array}{c} \mathbf{x} \\ \mathbf{x}' \\ \mathbf{w} \\ \mathbf{w}' \end{array} \right]^n \end{bmatrix} \approx \mathbf{b} \quad (5.118)$$

Nesse caso, \mathbf{x} é um vetor com os valores das variáveis (originais) de decisão x_i ; \mathbf{x}' é um vetor com os valores das variáveis (cópias) de decisão x_i^k ; \mathbf{w} é um vetor com os valores das variáveis (originais) de decisão w_{ij} ; \mathbf{w}' é um vetor com os valores das variáveis (cópias) de decisão w_{ij}^k ;

Relaxando as restrições presentes nas matrizes \mathbf{A} (5.91 e 5.92) no sentido lagrangiano, por meio do vetor de multiplicadores $\vec{\alpha}$, ($\vec{\alpha} \geq 0$), o problema $PQLT^n$ pode ser dividido em n subproblemas independentes. Considerando R_k como sendo as restrições associadas ao subproblema k , $k=1, \dots, n$, e d o número de restrições relaxadas, cada subproblema pode ser representado como descrito em 5.119, e a solução da relaxação do problema PQLT em n subproblemas (*clusters*) é dada pela Equação 5.120.

$$DC_\alpha PQLT_k : \quad v(DC_\alpha PQLT_k) = \mathbf{max}$$

$$\{ (\mathbf{q}_x^k - \vec{\alpha} \mathbf{A}_k) \mathbf{x}^k - (\vec{\alpha} \mathbf{A}_k) \mathbf{x}'^k + (\mathbf{q}_w^k - \vec{\alpha} \mathbf{A}_k) \mathbf{w}^k + (\mathbf{q}_{w'}^k - \vec{\alpha} \mathbf{A}_k) \mathbf{w}'^k \} \quad (5.119)$$

$$\mathbf{x}^k, \mathbf{x}'^k, \mathbf{w}^k, \mathbf{w}'^k \in R_k$$

$$DC_\alpha PQLT^n : v(DC_\alpha PQLT^n) = \sum_{k=1}^n v(DC_\alpha PQLT_k) - \sum_{p=1}^d \alpha_p \quad (5.120)$$

Logo, assim como descrito na seção anterior, pode-se gerar o seguinte PMR:

$$PMR_{PL}^{DC} : \quad v(PMR_{PL}^{DC}) = \mathbf{Maximizar}:$$

$$\sum_{k=1}^n \lambda_{k,t} \left(\mathbf{q}_x^{k,t} \mathbf{x}^{k,t} + \mathbf{0} \mathbf{x}'^{k,t} + \mathbf{q}_w^{k,t} \mathbf{w}^{k,t} + \mathbf{q}_{w'}^{k,t} \mathbf{w}'^{k,t} \right) \quad (5.121)$$

Sujeito a:

$$\sum_{k=1}^n \sum_{t \in T_k} \lambda_{k,t} (\mathbf{A}_k \mathbf{y}^{k,t}) \approx \mathbf{b}, \text{ onde } \mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}' \\ \mathbf{w} \\ \mathbf{w}' \end{bmatrix} \quad (5.122)$$

$$\sum_{t \in T_k} \lambda_{k,t} = 1 \quad \forall k \in \{1, \dots, n\} \quad (5.123)$$

$$\lambda_{k,t} \geq 0 \quad \forall k \in \{1, \dots, n\}, t \in T_k \quad (5.124)$$

Para cada subproblema k (5.119), pode-se substituir o vetor de multiplicadores lagrangianos $\vec{\alpha}$ pelo vetor de variáveis duais $\vec{\pi}$ associado com as restrições 5.122, e de uma forma alternativa, cada subproblema pode ser descrito pela Equação 5.125, e a relaxação do PQLT, em n clusters, pela Equação 5.126.

$$Z_k^{DC} = \mathbf{max} \left\{ \begin{array}{l} (\mathbf{q}_x^{k,t} - \vec{\pi} \mathbf{A}_k) \mathbf{x}^{k,t} + (\mathbf{q}_{x'}^{k,t} - \vec{\pi} \mathbf{A}_k) \mathbf{x}'^{k,t} \\ + (\mathbf{q}_w^{k,t} - \vec{\pi} \mathbf{A}_k) \mathbf{w}^{k,t} + (\mathbf{q}_{w'}^{k,t} - \vec{\pi} \mathbf{A}_k) \mathbf{w}'^{k,t} \\ \mathbf{x}^{k,t}, \mathbf{x}'^{k,t}, \mathbf{w}^{k,t}, \mathbf{w}'^{k,t} \in R_k \end{array} \right\} \quad (5.125)$$

$$DC_{\pi}PQLT^n : \quad v(DC_{\pi}PQLT^n) = \sum_{k=1}^n Z_k^{DC} + \sum_{p=1}^d \pi_p \quad (5.126)$$

O PMR inicial é gerado de forma análoga à apresentada na seção anterior (Figura 5.2), e os valores dados às variáveis “copiadas” são os mesmos obtidos para as variáveis originais. Novas colunas são geradas até que um critério de parada seja satisfeito, e o PMR final, formado por todas as colunas geradas, é resolvido de forma inteira ($\lambda_{k,t} \in \{0, 1\}$). Esse problema será tratado como PMR_{PLI}^{DC} , e o valor de sua solução como $v(PMR_{PLI}^{DC})$. A geração de colunas é realizada de forma semelhante à apresentada na Figura 5.3, porém, na linha 5, $v(LC_{\pi}PQL^n)$ é substituído por $v(DC_{\pi}PQLT^n)$ e $v(PMR_{PL}^{LC})$ por $v(PMR_{PL}^{DC})$, e na linha 7, $v(PMR_{PLI}^{LC})$ é substituído por $v(PMR_{PLI}^{DC})$. Um exemplo de aplicação dessa abordagem é apresentado na Subseção A.4.2.

5.8 Utilização de uma posiforme

Como definido em Hammer e Simeone (1989), uma “posiforme”, ou função pseudo-booleana de termos positivos, consiste em uma função na qual todas as variáveis possuem apenas coeficientes positivos. Werra e Hammer (2007) mostram que qualquer função pseudo-booleana g pode ser expressa como $g = M + f$, onde M é um valor constante e f uma posiforme. Assim, qualquer função pseudo-booleana g pode ser expressa, de várias maneiras, como uma constante mais uma posiforme qualquer.

Como apresentado na Seção 5.9, os métodos descritos anteriormente encontraram dificuldades na resolução das instâncias com matrizes \mathbf{Q} de alta densidade. Então, com o intuito de encontrar uma formulação que gere um grafo com menor número de conexões (arestas), o que consequentemente poderá gerar *clusters* mais “independentes”, a formulação original do PQ (Equação 5.1) pode ser transformada em uma posiforme por meio da eliminação dos termos quadráticos $x_i x_j$ com coeficiente $q_{ij} < 0$. Esses termos podem ser eliminados pela substituição das variáveis presentes nesses termos pelos seus complementos ($x_i = 1 - \bar{x}_i$; $x_j = 1 - \bar{x}_j$). Logo, para simplificar a transformação, cada termo quadrático $q_{ij} x_i x_j$, $\forall q_{ij} < 0$, é substituído pelo termo $q_{ij} + |q_{ij}| \bar{x}_i + |q_{ij}| x_i \bar{x}_j$. Assim, uma posiforme para o problema de programação

quadrática binária irrestrita (PPQ) pode ser escrita da seguinte maneira:

$$PPQ: \quad v(PPQ) = \max \sum_{q_{ij} \geq 0; i < j} q_{ij} x_i x_j + \sum_{q_{ij} < 0; i < j} q_{ij} + |q_{ij}| \bar{x}_i + |q_{ij}| x_i \bar{x}_j \quad (5.127)$$

Segundo Hammer e Simeone (1989), uma linearização interessante para esse modelo pode ser obtida pela formação de um grafo $G^P = (V^P, A^P)$, onde V^P é o conjunto de todos os termos da função, quadráticos ou não, e A^P é o conjunto de arestas onde $a_{ij}^P = 1$ se o complemento de uma variável presente no termo i estiver presente no termo j , ou vice-versa, e $a_{ij}^P = 0$ caso contrário. A Figura 5.4 apresenta um exemplo desse grafo.

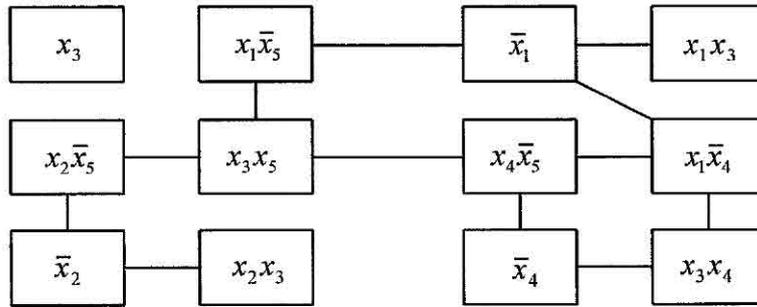


Figura 5.4 - Grafo gerado pelo modelo PPQ.

A partir desse grafo, cada vértice é substituído por uma variável y_i , $i = 1, \dots, t$, onde t é o número de vértices do grafo ($t = |V^P|$). Além disso, para garantir a equivalência com a formulação original do problema PQ, deve-se garantir que apenas um dos vértices adjacentes pertençam à solução do problema, ou seja, como a adjacência é formada pelos complementos das variáveis, apenas um dos vértices poderá fazer parte da solução. Logo, considerando c_i como o coeficiente do termo i (formado pelos coeficientes q_{ij} dos termos quadráticos ou não de PPQ), e M como o termo constante resultante do modelo posiforme (q_{ij} restante do segundo termo da função objetivo de PPQ), o PPQ pode ser reescrito de uma forma linear como descrito pelo modelo $PPQL$. Um exemplo da obtenção da posiforme é apresentado na Seção A.5.

$PPQL: \quad v(PPQL) = \text{Maximizar:}$

$$\sum_{i=1}^t c_i y_i + M \quad (5.128)$$

Sujeito a:

$$y_i + y_j \leq 1 \quad a_{ij} = 1; i < j; i, j = 1, \dots, t \quad (5.129)$$

$$y_i \in \{0, 1\} \quad i = 1, \dots, t \quad (5.130)$$

5.8.1 LagClus aplicada ao PPQL

Seguindo as idéias da LagClus apresentada na Seção 5.5, o grafo G^P pode ser particionando em n ($n \leq t$) *clusters* independentes, por meio das heurísticas disponíveis na biblioteca METIS (KARYPIS; KUMAR, 1998). Logo, tem-se $V^P = V_1^P \cup V_2^P \cup \dots \cup V_n^P$, onde $V_i^P \cap V_j^P = \emptyset$ ($\forall i, j = 1, \dots, n, i \neq j$), $G_i^P = (V_i^P, A_i^P)$, $i = 1, \dots, n$, e $X_i^P = V^P - V_i^P$, $i = 1, \dots, n$. A partir de então, o modelo PPQL pode ser reescrito da seguinte forma:

$PPQL^n$: $v(PPQL^n) = \text{Maximizar:}$

$$\sum_{k=1}^n \left(\sum_{i \in V_k^P} c_i y_i \right) + M \quad (5.131)$$

Sujeito a:

$$y_i + y_j \leq 1 \quad i \in V_k^P, j \in V_k^P, i < j, k = 1, \dots, n \quad (5.132)$$

$$y_i + y_j \leq 1 \quad i \in V_k^P, j \in X_k^P, i < j, k = 1, \dots, n \quad (5.133)$$

$$y_i \in \{0, 1\} \quad i \in V_k^P, k = 1, \dots, n \quad (5.134)$$

Nesse caso, a restrição 5.132 trata as arestas (i, j) cujos vértices são internos ao *cluster* (sub-grafo) k (arestas *intra-cluster*), e a restrição 5.133 trata as arestas (i, j) cujos vértices estão em *clusters* distintos (arestas *inter-clusters* ou arestas de ligação).

A restrição 5.133 pode ser relaxada no sentido lagrangiano por meio do vetor de multiplicadores $\vec{\alpha}$, gerando assim n subproblemas independentes, formulados da seguinte maneira:

$LC_\alpha PPQL_k$: $v(LC_\alpha PPQL_k) = \text{Maximizar:}$

$$\sum_{i \in V_k^P} c_i y_i - \sum_{\substack{i \in V_k^P; \\ j \in X_k^P}} \alpha_{ij} y_i - \sum_{\substack{i \in X_k^P; \\ j \in V_k^P}} \alpha_{ij} y_j \quad (5.135)$$

Sujeito a:

$$y_i + y_j \leq 1 \quad i \in V_k^P, j \in V_k^P, i < j \quad (5.136)$$

$$y_i \in \{0, 1\} \quad i \in V_k^P \quad (5.137)$$

Por fim, a solução da relaxação do problema $PPQL$ (que é equivalente a PQ) em n subproblemas (*clusters*) é dada pela Equação 5.138, e o seu dual lagrangiano é apresentado na Equação 5.139.

$$LC_\alpha PPQL^n : \quad v(LC_\alpha PPQL^n) = \sum_{k=1}^n v(LC_\alpha PPQL_k) + \sum_{\substack{\alpha_{ij}=1; \\ cl(i) \neq cl(j)}} \alpha_{ij} + M \quad (5.138)$$

$$DLC_\alpha PPQL^n : \quad v(DLC_\alpha PPQL^n) = \text{Min}\{v(LC_\alpha PPQL^n)\}, \alpha \geq 0 \quad (5.139)$$

Um exemplo de aplicação dessa abordagem é apresentado na Subseção A.5.1.

5.8.2 DecLag aplicada ao $PPQL$

A partir do grafo G^P descrito na seção anterior, o $PPQL$ pode ser decomposto em n subproblemas por meio da cópia dos vértices que formam as arestas *inter-clusters*. Logo, o $PPQL$ pode ser reescrito da seguinte forma:

$PPQL^n$: $v(PPQL^n) = \text{Maximizar:}$

$$\sum_{k=1}^n \left(\sum_{i \in V_k^P} c_i y_i \right) + M \quad (5.140)$$

Sujeito a:

$$y_i + y_j \leq 1 \quad i \in V_k^P, j \in V_k^P, i < j, k = 1, \dots, n \quad (5.141)$$

$$y_i + y_{j'}^k \leq 1 \quad i \in V_k^P, j \in X_k^P, i < j, k = 1, \dots, n \quad (5.142)$$

$$y_j = y_{j'}^k \quad j \in X_k^P, k = 1, \dots, n \quad (5.143)$$

$$y_i \in \{0, 1\} \quad i \in V_k^P, k = 1, \dots, n \quad (5.144)$$

Nesse modelo, as variáveis $y_{j'}^k$ representam as cópias do vértice j (j') no *cluster* k . A restrição 5.141 trata as arestas (i, j) cujos vértices são internos ao *cluster* (sub-grafo) k (arestas *intra-cluster*). Já a restrição 5.142 trata as arestas (i, j) cujos vértices estão em *clusters* distintos (arestas *inter-clusters* - arestas de ligação), e a restrição 5.143 é a restrição de cópia, que garante a igualdade entre as variáveis originais e suas cópias. Logo, relaxando a restrição de cópia (5.143) no sentido lagrangiano por meio do vetor de multiplicadores $\vec{\alpha}$, pode-se dividir o problema *PPQL* em n subproblemas independentes, formulados da seguinte maneira:

$DC_\alpha PPQL_k$: $v(DC_\alpha PPQL_k) = \text{Maximizar:}$

$$\sum_{i \in V_k^P} \left(c_i - \sum_{d \neq k} \alpha_i^d \right) y_i + \sum_{\substack{\exists \alpha_{ij}=1; \\ i \in V_k^P, j \in X_k^P}} \alpha_j^k y_{j'}^k \quad (5.145)$$

Sujeito a:

$$y_i + y_j \leq 1 \quad i \in V_k^P, j \in V_k^P, i < j \quad (5.146)$$

$$y_i + y_{j'}^k \leq 1 \quad i \in V_k^P, j \in X_k^P, i < j \quad (5.147)$$

$$y_j^k \in \{0, 1\} \quad j \in X_k^P \quad (5.148)$$

$$y_i \in \{0, 1\} \quad i \in V_k^P \quad (5.149)$$

Por fim, a solução da relaxação do problema *PPQL* em n subproblemas (*clusters*) é dada pela Equação 5.150, e o seu dual lagrangiano é apresentado na Equação 5.151.

$$DC_\alpha PPQL^n : \quad v(DC_\alpha PPQL^n) = \sum_{k=1}^n v(DC_\alpha PPQL_k) + M \quad (5.150)$$

$$DDC_\alpha PPQL^n : \quad v(DDC_\alpha PPQL^n) = \text{Min}\{v(DC_\alpha PPQL^n)\} \\ \alpha \text{ irrestrito} \quad (5.151)$$

Um exemplo de aplicação dessa abordagem é apresentado na Subseção A.5.2.

5.9 Experimentos computacionais

Vários experimentos computacionais foram realizados com a utilização de um conjunto de 45 instâncias disponíveis na *OR-Library* (BEASLEY, 1990). Essas instâncias

foram criadas pelo gerador proposto por Pardalos e Rodgers (1990a), e separadas em 6 classes (A, B, C, D, E e F) com diferentes características (m , densidade, e intervalo dos elementos da matriz \mathbf{Q}). Os problemas das classes A, B e C foram propostos por Pardalos e Rodgers (1990a), e os demais por Glover et al. (1998), que afirmam que, devido a suas características, essas instâncias estão entre as mais difíceis encontradas na literatura.

Para a divisão dos grafos G e G^P , foram utilizadas as heurísticas disponíveis na biblioteca METIS (KARYPIS; KUMAR, 1998), que segundo Hicks et al. (2005) apresentam bons resultados para o particionamento de grafos. Essas heurísticas dividem os grafos minimizando o número de arestas com terminações em *clusters* distintos.

Os duais lagrangianos das relaxações lagrangianas tradicionais ($DL_\alpha PQL$, $DL_\beta PQL$, etc) e da LagClus e DecLag, incluindo a posiforme ($DLC_{\alpha\beta\lambda} PQL^n$, $DLC_{\beta\lambda} PQL^n$, $DDC_{\alpha\beta} PQL^n$, $DDC_\alpha PQL^n$, $DDC_{\alpha\beta} PQLT^n$, $DLC_\alpha PPQL^n$, $DDC_\alpha PPQL^n$) foram otimizados pelo Algoritmo de Subgradientes descrito na Seção 2.9, com o limite de tempo de 1 hora de processamento.

Para resolver a relaxação linear do problema (*roof dual*) e a relaxação linear com a restrição de corte (PQLC) (ver Seção 5.4), foi utilizado o CPLEX 10.0 (ILOG, 2006). Este mesmo *software* foi utilizado para resolver os subproblemas de forma exata a cada iteração do Algoritmo de Subgradientes na resolução do dual lagrangiano para as relaxações lagrangianas tradicionais e para as abordagens propostas da LagClus e DecLag, incluindo a formulação posiforme (ver Seções 5.3, 5.5, 5.6 e 5.8). Nas abordagens baseadas na geração de colunas (ver Seção 5.7), os PMRs e os subproblemas também foram resolvidos pelo CPLEX. Em todos os casos foi utilizado o tempo limite de 1 hora de processamento.

A Tabela 5.1 apresenta os resultados obtidos pela resolução da relaxação linear de PQL (*roof dual*) e dessa mesma relaxação porém com a inserção da restrição de corte (PQLC) proposta na Seção 5.4. Nessa tabela, os *gaps* (%) são calculados de acordo com a Equação 5.152, onde $v(Opt)$ é o valor da melhor solução conhecida (GLOVER et al., 1998) para a respectiva instância, e $v(limitante)$ é o valor do limitante obtido pela resolução das relaxações lineares (com e sem a restrição de corte). A coluna *Id.* apresenta os “nomes” das instâncias; m é o número de vértices; *Dsde* é a densidade (%) da matriz \mathbf{Q} ; e *Cortes* é o número de restrições de cortes inseridas no modelo.

Tabela 5.1 - Resultados obtidos pelas relaxações lineares do PQL.

Instâncias				Roof dual		PQLC		
Id.	<i>m</i>	Dsde.	<i>v(Opt)</i>	<i>v(roof dual)</i>	<i>Gap</i>	Cortes	<i>v(PQLC)</i>	<i>Gap</i>
1a	50	10	3414	3414,00	0	16	3414,00	0
2a	60	10	6063	6063,00	0	20	6063,00	0
3a	70	10	6037	6446,00	6,77	39	6421,00	6,36
4a	80	10	8598	8606,50	0,10	73	8606,50	0,10
5a	50	20	5737	6662,00	16,12	127	6402,00	11,59
6a	30	40	3980	5109,50	28,38	243	4802,50	20,67
7a	30	50	4541	5835,50	28,51	455	5021,50	10,58
8a	100	6,25	11109	11109,00	0	34	11109,00	0
1b	20	100	133	391,00	193,98	1086	260,67	95,99
2b	30	100	121	518,50	328,51	3892	345,67	185,67
3b	40	100	118	624,50	429,24	9577	416,33	252,82
4b	50	100	129	822,50	537,60	18837	548,33	325,06
5b	60	100	150	949,00	532,67	33131	632,67	321,78
6b	70	100	146	1137,00	678,77	52921	758,00	419,18
7b	80	100	160	1260,00	687,50	79450	840,00	425,00
8b	90	100	145	1435,50	890,00	113815	957,00	560,00
9b	100	100	137	1656,50	1109,12	157124	1104,33	706,08
10b	125	10	154	1965,00	1175,97	307516	1310,00	750,65
1c	40	80	5058	8374,50	65,57	5134	6136,00	21,31
2c	50	60	6213	10176,50	63,79	4734	7798,00	25,51
3c	60	40	6665	9600,50	44,04	2120	8149,50	22,27
4c	70	30	7398	10107,00	36,62	1478	8880,00	20,03
5c	80	20	7362	8955,00	21,64	660	8566,00	16,35
6c	90	10	5824	5910,00	1,48	120	5841,50	0,30
7c	100	10	7225	7225,00	0	143	7225,00	0
1d	100	10	6333	7063,50	11,53	173	6856,50	8,27
2d	100	20	6579	12297,00	86,91	1438	10810,00	64,31
3d	100	30	9261	18053,50	94,94	3850	14639,50	58,08
4d	100	40	10727	25156,50	134,52	10637	18576,48	73,17
5d	100	50	11626	30732,00	164,34	18834	21630,10	86,05
6d	100	60	14207	37334,50	162,79	34099	25646,00	80,52
7d	100	70	14476	44171,50	205,14	53919	30468,67	110,48
8d	100	80	16352	50239,50	207,24	79665	33660,00	105,85
9d	100	90	15656	55130,00	252,13	109387	36753,33	134,76
10d	100	100	19102	63830,50	234,16	156561	44323,33	132,04
1e	200	10	16464	23752,50	44,27	1217	22269,50	35,26
2e	200	20	23395	51348,00	119,48	10028	43015,00	83,86
3e	200	30	25243	72883,00	188,73	33424	53183,38	110,69
4e	200	40	35594	102110,50	186,88	82353	70654,38	98,50
5e	200	50	35154	127756,00	263,42	159221	87478,67	148,84
1f	500	10	61194	158664,00	159,28	20339	138369,00	126,12
2f	500	25	100161	397839,50	297,20	317717	265003,00	164,58
3f	500	50	138035	793264,00	474,68	-	-	-
4f	500	75	172771	1188603,00	587,96	-	-	-
5f	500	100	190507	1143651,00	500,32	-	-	-
Tempo médio (seg.)				174,45		154,97		

Analisando a Tabela 5.1, percebe-se que a simples inserção da restrição de corte proposta na Seção 5.4 melhora significativamente a qualidade dos limitantes obtidos, e o mais interessante, ainda apresenta uma pequena redução no tempo de processamento. Entretanto, o CPLEX não foi capaz de obter soluções para as maiores instâncias (3f, 4f e 5f) em um tempo de 1 hora de processamento.

$$gap = \frac{v(\textit{limitante}) - v(\textit{Opt})}{v(\textit{Opt})} \times 100 \quad (5.152)$$

A Tabela 5.2 apresenta os *gaps* (%) obtidos pelas relaxações lagrangianas tradicionais (ver Seção 5.3). Nessa tabela, os *gaps* também são calculados de acordo com a Equação 5.152, porém $v(\textit{limitante})$ é o valor do limitante obtido pela resolução do dual lagrangiano referente ao respectivo modelo. As instâncias da classe B são formadas por matrizes \mathbf{Q} nas quais todos os elementos externos à diagonal principal são negativos ou nulos ($q_{ij} \leq 0, \forall i < j$). Logo, na Tabela 5.2, as relaxações lagrangianas baseadas nas arestas positivas ($L_\alpha PQL^n$, $L_\beta PQL^n$ e $L_{\alpha\beta} PQL^n$) não foram utilizadas. Os resultados apresentados para as demais relaxações são idênticos, pois as mesmas restrições foram relaxadas. A última linha dessa tabela apresenta o tempo médio de execução (em segundos) para cada instância. Os melhores *gaps* são destacados em negrito. Fica claro que as relaxações baseadas nas restrições referentes às arestas positivas são mais fortes, e apresentam melhores *gaps* para as instâncias utilizadas, porém em um tempo computacional mais elevado.

Na Tabela 5.3 são apresentados os resultados obtidos pelas abordagens baseadas na LagClus (ver Seção 5.5). Nessa tabela, os *gaps* também são calculados de acordo com a Equação 5.152, na qual $v(\textit{limitante})$ é o valor do limitante obtido pela resolução do dual lagrangiano referente ao respectivo modelo. A coluna % Cortes apresenta o percentual de arestas cortadas (arestas *inter-clusters*); %R. é o percentual de restrições relaxadas; e n é o número de *clusters* em que o problema é particionado. O valor de n foi determinado empiricamente, por meio de experimentos. Os melhores *gaps* são destacados em negrito. Fica claro que o percentual de restrições relaxadas é menor na 2ª abordagem ($LC_{\beta\lambda} PQL^n$), pois a restrição 5.15 não é relaxada, ao contrário da 1ª abordagem. Para as instâncias das classe A e B, os *gaps* obtidos são praticamente os mesmos. Entretanto, para as demais instâncias, pode-se observar uma significativa superioridade da 2ª abordagem em relação à 1ª. Além da melhor qualidade dos *gaps* obtidos, a 2ª abordagem ainda apresentou um menor tempo médio de processamento.

Tabela 5.2 - Gaps (%) obtidos pelas relaxações lagrangianas tradicionais.

Inst.	DL _α PQL	DL _β PQL	DL _λ PQL	DL _{α,β} PQL	DL _{αλ} PQL	DL _{βλ} PQL	DL _{α,βλ} PQL
1a	0	0	0	0	0	0	0
2a	0	0	0	0	0	0	0
3a	3,37	3,37	6,78	3,39	6,79	6,78	6,79
4a	0,10	0,10	0,10	0,10	0,10	0,10	0,25
5a	4,17	4,17	16,13	4,29	16,16	16,14	16,16
6a	8,48	8,48	28,38	8,49	28,49	28,46	28,47
7a	7,69	7,69	28,51	7,70	28,61	28,61	28,60
8a	0	0	0	0	0	0	0
1b	-	-	194,66	-	194,66	194,66	194,66
2b	-	-	329,68	-	329,68	329,68	329,68
3b	-	-	429,66	-	429,66	429,66	429,66
4b	-	-	538,65	-	538,65	538,65	538,65
5b	-	-	534,36	-	534,36	534,36	534,36
6b	-	-	680,98	-	680,98	680,98	680,98
7b	-	-	690,11	-	690,11	690,11	690,11
8b	-	-	892,05	-	892,05	892,05	892,05
9b	-	-	1109,76	-	1109,76	1109,76	1109,76
10b	-	-	1176,72	-	1176,72	1176,72	1176,72
1c	17,31	17,27	65,73	17,33	65,80	65,83	65,71
2c	20,43	20,43	63,95	21,21	64,02	64,01	64,04
3c	14,31	14,24	44,09	14,54	44,21	44,19	44,14
4c	9,11	9,15	36,70	9,49	36,76	36,75	36,73
5c	6,90	6,91	21,68	6,97	21,72	21,72	21,72
6c	0	0	1,48	0,06	1,48	1,48	1,48
7c	0	0	0	0	0	0	0,18
1d	1,73	1,75	11,54	1,82	11,57	11,57	11,56
2d	44,96	46,85	86,98	60,09	87,25	87,18	87,20
3d	57,69	60,07	95,01	79,38	95,28	95,25	95,14
4d	92,15	95,73	134,57	171,27	135,03	135,00	134,93
5d	125,90	112,79	164,38	203,18	164,93	164,94	164,67
6d	116,84	113,68	163,05	203,1	163,43	163,41	163,10
7d	162,80	162,54	205,16	241,26	205,93	205,90	205,52
8d	181,96	151,40	207,67	246,30	208,05	208,04	207,61
9d	194,06	226,05	252,45	284,30	253,10	253,06	252,62
10d	183,00	197,91	234,55	274,54	235,05	235,03	234,82
1e	28,34	28,26	44,41	48,97	44,43	44,43	44,43
2e	94,40	105,59	119,57	160,25	119,93	119,94	119,72
3e	144,66	155,59	188,80	225,73	189,42	189,45	189,09
4e	197,18	187,58	186,91	217,13	187,52	187,60	187,45
5e	228,72	258,77	263,81	298,72	264,43	264,44	264,08
1f	124,66	165,98	159,40	423,65	159,88	159,86	159,60
2f	306,49	301,88	297,34	696,82	298,33	298,36	297,77
3f	26,35	19,33	474,75	1050,44	476,44	476,44	476,02
4f	11,08	28,21	588,45	1276,14	590,24	590,18	588,98
5f	9,41	29,68	747,97	1557,97	731,40	731,38	730,04
Tempo	2617,25	2725,84	298,46	2422,59	18,05	19,99	16,05

Tabela 5.3 - Resultados obtidos pelas abordagens baseadas na LagClus.

Inst.	n	%	DLC _{αβλ} PQL ⁿ			DLC _{βλ} PQL ⁿ		
			%R.	v(DLC _{αβλ} PQL ⁿ)	Gap	%R.	v(DLC _{βλ} PQL ⁿ)	Gap
1a	2	12,82	14,15	3414,00	0	9,43	3414,00	0
2a	2	19,37	18,71	6063,00	0	13,19	6063,00	0
3a	2	20,55	21,08	6056,02	0,32	13,45	6056,02	0,32
4a	2	22,72	21,38	8606,50	0,10	14,31	8606,50	0,10
5a	2	25,98	23,38	5757,44	0,36	15,80	5757,52	0,36
6a	2	31,37	27,01	3985,51	0,14	18,39	3985,50	0,14
7a	2	34,44	29,86	4541,00	0	19,67	4541,00	0
8a	2	18,11	18,59	11109,00	0	12,01	11109,00	0
1b	2	46,38	25,67	184,50	38,72	25,67	184,50	38,72
2b	2	47,49	25,41	181,00	49,59	25,41	181,00	49,59
3b	2	47,84	25,13	180	52,54	25,13	180	52,54
4b	2	48,33	25,14	190,50	47,68	25,14	190,50	47,68
5b	2	48,43	25,04	208,49	38,99	25,04	208,49	38,99
6b	2	48,70	25,06	222,00	52,05	25,06	222,00	52,05
7b	2	48,86	25,06	201,50	25,93	25,06	201,50	25,93
8b	2	48,96	25,03	241,00	66,21	25,03	241,00	66,21
9b	2	49,04	25,02	239,50	74,82	25,02	239,50	74,82
10b	2	49,10	24,94	252,34	63,86	24,94	252,34	63,86
1c	2	43,91	35,36	5750,10	13,68	23,36	5750,60	13,69
2c	2	41,21	33,09	6844,50	10,16	21,95	6844,72	10,17
3c	2	36,01	29,32	6665,00	0	19,54	6665,00	0
4c	2	34,73	28,06	7443,55	0,62	19,03	7439,00	0,55
5c	2	31,35	27,22	7395,05	0,45	17,63	7389,41	0,37
6c	2	21,88	21,25	5824,00	0	13,38	5824,00	0
7c	2	25,88	23,23	7225,10	0	15,56	7225,00	0
1d	2	24,28	22,67	6333,00	0	14,57	6333,00	0
2d	2	33,96	28,69	8349,57	26,91	18,65	8214,67	24,86
3d	2	36,98	29,89	12575,85	35,79	19,79	11787,68	27,28
4d	2	39,90	31,03	19465,91	81,47	20,95	17637,74	64,42
5d	2	41,11	32,19	24829,63	113,57	21,40	21820,62	87,69
6d	2	43,47	33,73	29730,78	109,27	22,47	25897,54	82,29
7d	2	44,76	34,38	40696,84	181,13	23,02	31540,50	117,88
8d	2	46,04	34,96	40846,66	149,80	23,61	35915,12	119,64
9d	2	47,02	35,80	48352,15	208,84	24,05	40877,94	161,10
10d	2	48,98	37,49	77364,00	305,00	24,98	47760,45	150,03
1e	3	45,15	37,06	18731,79	13,77	24,92	18653,02	13,30
2e	4	59,85	47,30	42846,40	83,14	31,45	42013,66	79,58
3e	4	63,91	49,27	63405,93	151,18	33,03	59811,36	136,94
4e	4	66,34	50,81	101943,95	186,41	34,01	84919,40	138,58
5e	4	68,18	52,48	133534,20	279,85	34,77	114078,76	224,51
1f	6	69,31	53,80	172822,27	182,42	36,04	142519,50	132,90
2f	10	82,66	63,04	375201,58	274,60	41,99	373017,43	272,42
3f	10	86,18	65,10	1426487,00	933,42	43,43	756776,14	448,25
4f	10	87,98	66,37	2161768,00	1151,23	44,23	1139747,00	559,69
5f	10	89,72	67,58	2907009,00	1425,93	45,04	1536025,64	706,28
Tempo médio				2789,82			2170,53	

Os resultados obtidos pelas abordagens baseadas na DecLag (ver Seção 5.6) são apresentados na Tabela 5.4. Mais uma vez, os *gaps* são calculados de acordo com a Equação 5.152, na qual $v(\textit{limitante})$ é o valor do limitante obtido pela resolução do dual lagrangiano referente ao respectivo modelo. A coluna *%C.* apresenta o percentual de variáveis (arestas e vértices) copiadas. O valor de n foi determinado empiricamente, por meio de experimentos. Os melhores *gaps* são destacados em negrito.

Com o intuito de comprovar a qualidade dos particionamentos realizados pela METIS (KARYPIS; KUMAR, 1998), a 3ª abordagem foi executada sobre um grafo particionado aleatoriamente ($DC_{\alpha\beta}PQLT^{n*}$ - ver Tabela 5.4), assim como apresentado em Billionnet et al. (1999) e Chardaire e Sutter (1995).

Analisando a Tabela 5.4, percebe-se claramente a influência positiva da utilização da METIS, pois o percentual de variáveis copiadas sem a utilização da METIS é expressivamente maior. Além disso, vale ressaltar que a 1ª e a 2ª abordagens apresentam um percentual de cópias bem inferior, o que é justificado pelo fato da 3ª abordagem trabalhar com todos os elementos da matriz de coeficientes \mathbf{Q} .

Na Tabela 5.4, a relaxação $DC_{\alpha\beta}PQLT^n$ é mais forte que as demais, e os *gaps* apresentados de melhor qualidade. Para as instâncias da classe F, todas as decomposições apresentaram resultados apenas para a instância 1f (com 10% de densidade). Para as demais instâncias, foram testados vários números de *clusters* (n), e em nenhum caso foram obtidas soluções em um tempo aproximado de 1 hora de processamento. Os tempos médios de processamento apresentados são semelhantes entre as abordagens propostas.

A Tabela 5.5 apresenta os *gaps* obtidos pelas abordagens baseadas na geração de colunas (ver Seção 5.7). Nesses casos, os *gaps* obtidos pelas soluções encontradas (PMR_{PLI}^{DC} e PMR_{PLI}^{LC}) foram praticamente os mesmos tanto para a Relaxação Lagrangiana com *Clusters* (LagClus) quanto para a Decomposição Lagrangiana (DecLag). As soluções apresentadas pela abordagem baseada na LagClus foram ligeiramente melhores. Os *gaps* apresentados para as soluções encontradas (PMR_{PLI}^{LC} e PMR_{PLI}^{DC}) foram calculados de acordo com a Equação 5.153. Já os *gaps* apresentados para os limitantes foram calculados de acordo com a Equação 5.152.

Tabela 5.4 - Resultados obtidos pelas abordagens baseadas na Declag.

Inst.	n	DDC _{αβ} PQL ⁿ		DDC _α PQL ⁿ		DDC _{αβ} PQLT ⁿ		DDC _{αβ} PQLT ^{n*}	
		%C.	Gap	%C.	Gap	%C.	Gap	%C.	Gap
1a	2	23,08	0	10,26	0,14	29,49	0	62,18	0
2a	2	34,08	0	14,80	0	41,70	0	63,23	0
3a	2	32,76	0,31	12,29	0,32	39,93	0	63,82	0,31
4a	2	35,42	0	12,76	0	42,45	0	59,11	0
5a	2	38,08	0	12,10	0	42,70	0	56,94	0
6a	2	43,14	0	11,76	0	46,08	0	60,78	0
7a	2	44,81	0	10,37	0	46,89	0	57,26	0
8a	2	29,70	0	11,63	0	37,38	0	58,91	0
1b	2	52,66	15,66	6,28	15,65	56,04	0	57,97	0
2b	2	51,85	21,61	4,36	21,60	54,03	6,21	54,90	7,66
3b	2	51,85	29,48	4,06	29,49	52,71	8,24	53,69	22,06
4b	2	52,10	36,97	3,81	36,96	52,26	4,82	52,74	16,15
5b	2	51,46	21,19	3,04	21,19	51,74	3,31	52,57	7,88
6b	2	51,51	29,43	2,81	29,42	51,55	9,61	52,28	12,00
7b	2	51,26	19,84	2,40	19,87	51,36	0	51,83	5,03
8b	2	51,05	35,18	2,10	35,20	51,17	20,79	51,71	22,05
9b	2	50,99	49,41	1,96	49,44	51,03	27,71	51,37	27,97
10b	2	50,65	39,76	1,57	39,75	50,69	23,14	51,20	33,46
1c	2	49,47	0	5,56	0	49,92	0	53,23	0
2c	2	46,86	0	5,66	0	47,36	0	53,26	0
3c	2	43,10	0	7,10	0	43,89	0	54,66	0
4c	2	42,03	0	7,34	0	43,42	0	53,42	0
5c	2	40,92	0	9,57	0	42,44	0	55,06	0
6c	2	32,45	0	10,61	0	38,57	0	57,55	0
7c	2	38,32	0	12,44	0	42,52	0	59,83	0
1d	2	35,52	0	11,28	0	40,24	0	62,12	0
2d	2	41,40	15,48	7,44	5,74	42,92	5,47	55,73	8,57
3d	2	42,69	41,69	5,70	9,07	43,54	6,47	54,89	13,23
4d	2	44,14	92,47	4,24	21,34	44,67	9,67	53,33	15,16
5d	2	44,91	111,53	3,82	41,67	45,07	10,2	52,74	16,62
6d	2	46,72	118,33	3,25	27,81	46,75	1,57	51,80	11,23
7d	2	47,43	143,54	2,69	35,02	47,57	3,92	51,27	4,94
8d	2	48,52	135,74	2,47	20,24	48,54	4,32	49,19	6,13
9d	2	49,19	167,14	2,18	28,93	49,26	1,26	51,75	2,05
10d	2	50,93	161,87	1,96	25,61	50,97	4,29	51,47	5,30
1e	4	72,22	13,05	21,14	5,65	79,05	7,17	95,29	8,92
2e	5	81,88	149,98	16,69	60,42	84,57	51,75	95,74	58,41
3e	8	98,56	88,61	20,03	72,61	101,65	85,83	108,36	88,39
4e	10	102,53	94,40	19,44	73,55	105,26	93,33	110,61	94,29
5e	15	114,75	118,53	24,33	134,41	118,26	154,52	119,56	159,06
1f	20	128,64	95,68	45,20	93,44	150,46	92,52	159,32	98,46
2f	20	-	-	-	-	-	-	-	-
3f	20	-	-	-	-	-	-	-	-
4f	20	-	-	-	-	-	-	-	-
5f	20	-	-	-	-	-	-	-	-
Tempo		2044,23		1838,67		2313,50		2333,44	

$$gap = \frac{v(Opt) - v(PMR_{PLI}^{LCouDC})}{v(Opt)} \times 100 \quad (5.153)$$

Ainda na Tabela 5.5, a abordagem baseada na DecLag não obteve resultados para os maiores problemas devido à falta de memória no CPLEX. Entretanto, o tempo de processamento foi expressivamente menor. Percebe-se claramente que a abordagem baseada na DecLag encontra grandes dificuldades nos problemas com matrizes mais densas, o que faz sentido, pois nesses casos muitas variáveis são copiadas, e consequentemente os subproblemas ficam mais “pesados” para o CPLEX. É interessante notar também que os *gaps* apresentados pelos limitantes da DecLag (PMR_{PL}^{DC} e $DC_{\pi}PQLT^n$) foram melhores do que os apresentados pelos limitantes da LagClus (PMR_{PL}^{LC} e $LC_{\pi}PQL^n$).

A Tabela 5.6 apresenta os *gaps* obtidos pelas abordagens aplicadas ao modelo baseado na posiforme proposta. Nesses casos, ainda foi verificada a influência do particionamento do grafo considerando os pesos dos vértices (particionar o grafo visando minimizar a diferença entre a soma dos pesos dos vértices presentes em cada *cluster*). Logo, as colunas $DLC_{\alpha}PPQL^n$ e $DDC_{\alpha}PPQL^n$ indicam os modelos cujos grafos foram particionados sem considerar os pesos dos vértices, e as colunas $DLC_{\alpha}PPQL^n*$ e $DDC_{\alpha}PPQL^n*$ apresentam os *gaps* obtidos a partir dos modelos cujos grafos foram particionados considerando os pesos dos vértices.

De uma forma geral, a DecLag, sem considerar os pesos dos vértices no particionamento do grafo, apresentou os melhores resultados. Entretanto, esse método não foi capaz de resolver as maiores instâncias, e os resultados apresentados para as instâncias com alta densidade foram piores que os apresentados pela LagClus, e alguns casos até mesmo pelo *roof dual* (relaxação linear do modelo $PPQL$, no caso da posiforme). Os *gaps* apresentados pelos modelos cujos grafos foram particionados considerando os pesos dos vértices, assim como os tempos de processamento, foram piores em relação ao particionamento considerando apenas o número de vértices.

Como pode ser observado na Tabela 5.6, os *gaps* obtidos pelo modelo $PPQL$ são piores que os obtidos pelo PQL (ver Tabelas 5.3 e 5.4). Esse fato pode ser justificado pelos valores apresentados na coluna *V.D.*, que são extremamente pequenos, o que aparentemente indica que, mesmo com baixa densidade, o grafo referente ao $PPQL$ ainda é “muito conectado”, e sendo assim, os particionamentos continuam cortando um número relativamente grande de arestas.

Tabela 5.5 - Gaps (%) obtidos pelas abordagens baseadas na GC.

Inst.	GC para a LagClus			GC para a DecLag		
	PMR _{PL} ^{LC}	LC _τ PQL ⁿ	PMR _{PLI} ^{LC}	PMR _{PL} ^{DC}	DC _τ PQLT ⁿ	PMR _{PLI} ^{DC}
1a	0	12,92	0	0	1,67	0
2a	0	3,10	0	0	2,96	0
3a	0,31	24,47	0,03	0,03	1,50	0,03
4a	0,10	20,47	0,19	1,34	3,73	1,34
5a	0,36	26,41	0	0	0,90	0
6a	0,14	36,71	0	0	0,05	0
7a	0	46,94	0	0	5,59	0
8a	0	7,27	0	0	1,37	0
1b	38,72	38,72	0	0	15,04	0
2b	49,59	49,59	0	6,16	49,82	0
3b	52,54	52,54	0	7,95	98,64	0
4b	47,67	47,67	0	0,25	149,50	0
5b	39,00	39,00	0	2,89	300,80	0
6b	52,05	52,05	8,90	9,49	313,90	12,33
7b	25,94	25,94	0	0	330,73	0
8b	66,21	66,21	0	0,34	477,73	0
9b	74,82	74,82	1,46	2,76	634,98	2,92
10b	63,85	63,85	4,55	4,55	866,06	4,55
1c	12,67	81,50	0	0	2,81	0
2c	9,43	59,75	0	0	1,85	0
3c	0	56,25	0	0	1,53	0
4c	0,48	49,46	0	0	4,35	0
5c	0,35	40,65	0	0	4,65	0
6c	0	13,95	0	0	1,29	0
7c	0	17,69	0	0,19	7,40	0,19
1d	0,04	23,74	0,09	0,54	8,04	0,54
2d	18,58	93,53	0	0	26,58	0
3d	6,25	137,30	0	0	40,37	0
4d	16,50	203,49	0	0	74,65	0
5d	24,86	267,01	0,19	0,19	140,18	0,19
6d	23,96	276,48	0,35	0,35	83,61	0,35
7d	50,10	351,07	0	0	176,94	0
8d	49,13	329,91	0	0	219,90	0
9d	66,32	391,41	0,06	0,06	126,10	0,06
10d	74,88	360,71	0	0	202,35	0
1e	2,02	87,48	0,10	0,10	157,35	0,10
2e	27,92	257,81	0,08	0,08	622,24	0,08
3e	66,72	391,76	0,06	0,06	1009,46	0,06
4e	72,78	435,23	0	0	696,43	0
5e	122,26	544,77	0,01	0,01	1080,96	0,01
1f	0,38	493,14	0,38	0,38	1102,40	0,38
2f	0,22	715,30	0,32	-	-	-
3f	0,23	1468,73	0,23	-	-	-
4f	-	-	-	-	-	-
5f	-	-	-	-	-	-
Tempo		1983,96			1282,81	

Tabela 5.6 - Gaps (%) obtidos pelas abordagens baseadas na formulação posiforme.

Inst.	<i>roof dual</i>	DLC _α PPQL ⁿ	DLC _α PPQL ^{n*}	DDC _α PPQL ⁿ	DDC _α PPQL ^{n*}
1a	0	0,03	0,03	0	0
2a	0	0,01	0,02	0	0
3a	6,77	0,38	0,89	0	0,47
4a	0,10	0,10	0,10	0,01	0,10
5a	16,12	0,02	0,36	0,01	0,36
6a	28,38	0,66	0,06	0	0
7a	28,51	0,58	0,16	0	0,01
8a	0	0,01	0,01	0	0
1b	193,98	28,81	34,03	10,52	7,07
2b	328,51	47,50	80,31	28,13	38,94
3b	429,24	99,69	70,18	40,20	37,05
4b	537,60	142,80	142,06	37,69	45,44
5b	532,67	199,25	294,60	128,39	199,07
6b	678,77	352,29	1459,87	348,15	518,17
7b	687,50	776,82	754,89	647,01	11813,21
8b	890,00	930,76	3029,08	2098,22	7165,07
9b	1109,12	1348,52	1857,51	4920,78	16263,41
10b	1175,97	2350,88	15656,99	28075,72	55945,19
1c	65,57	9,28	13,81	0	0,65
2c	63,79	10,20	12,88	0	16,62
3c	44,04	1,72	2,79	0,01	0
4c	36,62	2,57	2,13	0	0,36
5c	21,64	0,80	0,41	0,01	0
6c	1,48	0,02	0,02	0,01	0
7c	0	0,01	0,03	0	0,01
1d	11,53	0,12	0,02	0,01	0,01
2d	86,91	37,71	39,97	19,89	20,73
3d	94,94	56,43	56,15	36,32	34,28
4d	134,52	93,07	95,01	55,22	61,57
5d	164,34	120,64	119,77	74,65	103,45
6d	162,79	131,15	136,20	99,95	125,66
7d	205,14	166,36	201,80	142,45	336,87
8d	207,24	168,79	181,34	255,48	244,91
9d	252,13	194,94	241,35	312,68	366,41
10d	234,16	181,78	217,19	273,53	306,80
1e	44,27	22,49	30,67	15,81	15,96
2e	119,48	98,66	98,08	84,28	90,05
3e	188,73	165,05	164,78	170,80	181,20
4e	186,88	175,00	175,81	239,34	243,86
5e	263,42	252,22	253,33	305,63	347,65
1f	159,28	149,78	149,83	119,71	130,60
2f	297,20	297,07	298,16	-	-
3f	-	543,95	584,88	-	-
4f	-	-	-	-	-
5f	-	-	-	-	-
Tempo	11,32	2585,13	2812,15	2675,02	3305,80

Na Tabela 5.7 é apresentada uma comparação entre o modelo linear do PQ (PQL) e o modelo baseado na linearização da posiforme PPQ ($PPQL$). Nessa tabela, a coluna $V.D.$ indica o número de vértices desconectados, ou seja, vértices que não possuem ligações no grafo, e conseqüentemente poderão ser eliminados do grafo, e seu coeficiente somado à constante M da função objetivo.

Pode-se notar que o número de vértices (m) e de arestas aumenta expressivamente no modelo $PPQL$, o que impossibilita a resolução das maiores instâncias. Entretanto, a densidade das matrizes de coeficientes diminui de forma ainda mais expressiva, o que em princípio é uma boa característica para os métodos apresentados neste capítulo.

As Tabelas 5.8 e 5.9 apresentam uma comparação entre os melhores limitantes obtidos pelos métodos descritos neste capítulo. Na Tabela 5.8 são apresentados os *gaps* obtidos para as instâncias das classes A, B e C, e na Tabela 5.9 para as instâncias das classes D, E e F. Nessas tabelas, a coluna *Lag* apresenta os melhores *gaps* obtidos pelas relaxações lagrangianas tradicionais (ver Seção 5.3). Analisando os *gaps* obtidos pelo *roof dual* tanto para o modelo PQL quanto o $PPQL$ (posiforme), nota-se que os valores são idênticos, porém para o $PPQL$ o CPLEX não foi capaz de resolver as maiores instâncias (devido ao grande número de vértices e arestas gerados nesse modelo - ver Tabela 5.7).

Para as instâncias das classes A e C (Tabela 5.8), os *gaps* obtidos pela terceira abordagem da DecLag ($DDC_{\alpha\beta}PQLT^n$) e pela relaxação linear do PMR da GC para a LagClus (PMR_{PL}^{LC}) foram praticamente os mesmos, e próximos a zero, confirmando a otimalidade das soluções já conhecidas. Entretanto, para as instâncias da classe B, os *gaps* apresentados pelo PMR_{PL}^{LC} foram melhores.

Os limitantes baseados nos subproblemas da GC ($LC_{\pi}PQL^n$ e $DC_{\pi}PQLT^n$) apresentaram altos *gaps* para praticamente todas as instâncias, o que provavelmente é justificado pelo grande *gap* de dualidade, como afirmado por Glover et al. (1998), e principalmente pelo fato dos métodos encontrarem rapidamente as soluções ótimas para o PMR, não gerando assim um número maior de colunas e conseqüentemente não permitindo a diminuição desses limitantes.

Tabela 5.7 - Comparação entre os modelos PQL e PPQL.

Inst.	PQL			PPQL			
	<i>m</i>	Arestas	Dsde.	<i>m</i>	Arestas	Dsde.	V.D.
1a	50	156	10	170	350	2,41	4
2a	60	222	10	238	563	1,98	0
3a	70	292	10	317	843	1,67	2
4a	80	383	10	407	1239	1,49	1
5a	50	281	20	306	1136	2,42	1
6a	30	204	40	219	939	3,90	0
7a	30	241	50	256	1320	4,01	0
8a	100	403	6,25	431	1105	1,19	4
1b	20	207	100	226	1488	5,80	0
2b	30	459	100	488	4850	4,06	0
3b	40	811	100	850	11265	3,11	0
4b	50	1258	100	1307	21617	2,53	0
5b	60	1811	100	1870	37011	2,12	0
6b	70	2458	100	2527	58316	1,83	0
7b	80	3205	100	3284	86696	1,61	0
8b	90	4052	100	4141	122812	1,43	0
9b	100	5002	100	5101	168814	1,30	0
10b	125	7788	10	7912	326659	1,04	0
1c	40	665	80	687	7571	3,20	0
2c	50	813	60	834	8756	2,51	0
3c	60	761	40	783	6399	2,08	0
4c	70	789	30	833	5931	1,71	0
5c	80	721	20	751	4183	1,48	0
6c	90	489	10	522	1825	1,34	0
7c	100	595	10	637	2433	1,20	0
1d	100	593	10	630	2404	1,21	2
2d	100	1116	20	1157	8553	1,28	0
3d	100	1525	30	1575	16148	1,30	0
4d	100	2100	40	2154	30146	1,30	0
5d	100	2513	50	2554	42353	1,30	0
6d	100	3048	60	3096	62686	1,31	0
7d	100	3532	70	3587	83891	1,30	0
8d	100	4007	80	4054	107059	1,30	0
9d	100	4445	90	4499	132584	1,31	0
10d	100	4996	100	5051	164906	1,29	0
1e	200	2124	10	2220	15670	0,64	0
2e	200	4127	20	4223	57977	0,65	0
3e	200	6048	30	6145	123409	0,65	0
4e	200	8117	40	8229	218381	0,64	0
5e	200	10055	50	10149	336759	0,65	0
1f	500	12917	10	13172	226237	0,26	0
2f	500	31514	25	31750	1328961	0,26	0
3f	500	62400	50	62645	5211985	0,27	0
4f	500	93248	75	93506	11597728	0,27	0
5f	500	124019	100	124270	20483074	0,27	0
Média	136,11	9255,78	50,58	9328,07	914111,82	1,66	0,31

Tabela 5.8 - Comparação entre os gaps (%) para as instâncias das classe A, B e C.

Inst.	PQL		Lag	LagClus		DecLag		GC + LagClus		GC + DecLag		Posiforme	
	roof	PQLC		DLC _α PQL ^A	DLC _α PQL ^A	DDC _{α,β} PQLT ^B	PMP _β L	LC _γ PQL ^A	PMP _β L	DC _γ PQLT ^B	roof	DLC _α PPQL ^A	DDC _α PPQL ^A
1a	0	0	0	0	0	0	12,92	0	1,67	0	0,03	0	
2a	0	0	0	0	0	0	3,10	0	2,96	0	0,01	0	
3a	6,77	6,36	3,37	0,32	0	0,31	24,47	0,03	1,50	6,77	0,38	0	
4a	0,10	0,10	0,10	0,10	0	0,10	20,47	1,34	3,73	0,10	0,10	0,01	
5a	16,12	11,59	4,17	0,36	0	0,36	26,41	0	0,90	16,12	0,02	0,01	
6a	28,38	20,67	8,48	0,14	0	0,14	36,71	0	0,05	28,38	0,66	0	
7a	28,51	10,58	7,69	0	0	0	46,94	0	5,59	28,51	0,58	0	
8a	0	0	0	0	0	0	7,27	0	1,37	0	0,01	0	
1b	193,98	95,99	194,66	38,72	0	38,72	38,72	0	15,04	193,98	28,81	10,52	
2b	328,51	185,67	329,68	49,59	6,21	49,59	49,59	6,16	49,82	328,51	47,50	28,13	
3b	429,24	252,82	429,66	52,54	8,24	52,54	52,54	7,95	98,64	429,24	99,69	40,20	
4b	537,60	325,06	538,65	47,68	4,82	47,67	47,67	0,25	149,50	537,60	142,80	37,69	
5b	532,67	321,78	534,36	38,99	3,31	39,00	39,00	2,89	300,80	532,67	199,25	128,39	
6b	678,77	419,18	680,98	52,05	9,61	52,05	52,05	9,49	313,90	678,77	352,29	348,15	
7b	687,50	425,00	690,11	25,93	0	25,94	25,94	0	330,73	687,50	776,82	647,01	
8b	890,00	560,00	892,05	66,21	20,79	66,21	66,21	0,34	477,73	890,00	930,76	2098,22	
9b	1109,12	706,08	1109,76	74,82	27,71	74,82	74,82	2,76	634,98	1109,12	1348,52	4920,78	
10b	1175,97	750,65	1176,72	63,86	23,14	63,85	63,85	4,55	866,06	1175,97	2350,88	28075,72	
1c	65,57	21,31	17,27	13,69	0	12,67	81,50	0	2,81	65,57	9,28	0	
2c	63,79	25,51	20,43	10,17	0	9,43	59,75	0	1,85	63,79	10,20	0	
3c	44,04	22,27	14,24	0	0	0	56,25	0	1,53	44,04	1,72	0,01	
4c	36,62	20,03	9,11	0,55	0	0,48	49,46	0	4,35	36,62	2,57	0	
5c	21,64	16,35	6,9	0,37	0	0,35	40,65	0	4,65	21,64	0,80	0,01	
6c	1,48	0,30	0	0	0	0	13,95	0	1,29	1,48	0,02	0,01	
7c	0	0	0	0	0	0	17,69	0,19	7,40	0	0,01	0	
Méd.	275,06	167,89	266,74	21,44	4,15	21,37	40,32	1,44	131,15	275,06	252,15	1453,39	

Tabela 5.9 - Comparação entre os gaps (%) para as instâncias da classe D, E e F.

Inst.	PQL		Lag	LagClus		DecLag	GC + LagClus		GC + DecLag		Posiforme		
	roof	PQLC		DLC _α PQL ⁿ	PMR _β ^C		LC _π PQL ⁿ	PMR _β ^{DC}	DC _π PQLT ⁿ	roof	DLC _α PPQL ⁿ	DDC _α PPQL ⁿ	
1d	11,53	8,27	1,73	0	0,04	0	23,74	0,54	8,04	11,53	0,12	0,01	
2d	86,91	64,31	44,96	24,86	18,58	5,47	93,53	0	26,58	86,91	37,71	19,89	
3d	94,94	58,08	57,69	27,28	6,25	6,47	137,30	0	40,37	94,94	56,43	36,32	
4d	134,52	73,17	92,15	64,42	16,50	9,67	208,49	0	74,65	134,52	93,07	55,22	
5d	164,34	86,05	112,79	87,69	24,86	10,20	267,01	0,19	140,18	164,34	120,64	74,65	
6d	162,79	80,52	113,68	82,29	23,96	1,57	276,48	0,35	83,61	162,79	131,15	99,95	
7d	205,14	110,48	162,54	117,88	50,10	3,92	351,07	0	176,94	205,14	166,36	142,45	
8d	207,24	105,85	151,40	119,64	49,13	4,32	329,91	0	219,90	207,24	168,79	255,48	
9d	252,13	134,76	194,06	161,10	66,32	1,26	391,41	0,06	126,10	252,13	194,94	312,68	
10d	234,16	132,04	183,00	150,03	74,88	4,29	360,71	0	202,35	234,16	181,78	273,53	
1e	44,27	35,26	28,26	13,30	2,02	7,17	87,48	0,10	157,35	44,27	22,49	15,81	
2e	119,48	83,86	94,40	79,58	27,92	51,75	257,81	0,08	622,24	119,48	98,66	84,28	
3e	188,73	110,69	144,66	136,94	66,72	85,83	391,76	0,06	1009,46	188,73	165,05	170,80	
4e	186,88	98,50	187,58	138,58	72,78	93,33	435,23	0	696,43	186,88	175,00	239,34	
5e	263,42	148,84	228,72	224,51	122,26	154,52	544,77	0,01	1080,96	263,42	252,22	305,63	
1f	159,28	126,12	124,66	132,90	0,38	92,52	493,14	0,38	1102,40	159,28	149,78	119,71	
2f	297,20	164,58	297,34	272,42	0,22	-	715,30	-	-	297,20	297,07	-	
3f	474,68	-	19,33	448,25	0,23	-	1468,73	-	-	-	543,95	-	
4f	587,96	-	11,08	559,69	-	-	-	-	-	-	-	-	
5f	500,32	-	9,41	706,28	-	-	-	-	-	-	-	-	
Méd.	218,80	95,38	112,97	177,38	34,62	33,27	379,38	0,11	360,47	165,47	158,62	137,86	

Ainda considerando as instâncias das classes A, B e C (Tabela 5.8), percebe-se claramente que todos os limitantes baseados na LagClus, na DecLag, e na GC foram expressivamente melhores que os apresentados pelas relaxações lagrangianas tradicionais e pela relaxação linear com a restrição de corte (PQLC), e o mais interessante, todos foram bem distantes do *roof dual*. Esse fato só não foi confirmado para as abordagens aplicadas ao modelo *PPQL* (posiforme) para as instâncias mais densas da classe B, o que é justificado pelo fato dessas instâncias serem formadas apenas por arestas negativas, o que conseqüentemente gera modelos de difícil solução, com um grande número de vértices e restrições (ver Tabela 5.7).

Para as instâncias das classes D e E (Tabela 5.9), os limitantes baseados na LagClus, na DecLag, e na GC, também foram melhores que os apresentados pelas relaxações lagrangianas tradicionais e pelo PQLC, e também apresentaram *gaps* bem distantes do *roof dual*. Já os limitantes baseados no *PPQL* apresentaram novamente dificuldades na resolução das instâncias mais densas.

Para as instâncias da classe F (Tabela 5.9), os métodos baseados na LagClus, DecLag e GC, além do PQLC, foram capazes de resolver apenas as instâncias com menor densidade, mas mesmo assim os resultados obtidos foram de boa qualidade.

De uma forma geral, analisando as Tabelas 5.8 e 5.9, percebe-se que os métodos baseados na DecLag encontram grandes dificuldades na resolução das instâncias que apresentam maior densidade, o que é justificado pelo grande número de variáveis que provavelmente deverão ser copiadas nos subproblemas, aumentando assim o seu tamanho e conseqüentemente dificultando sua resolução. Já as abordagens baseadas na geração de colunas (principalmente com a DecLag) apresentaram os melhores *gaps* entre todas as abordagens propostas, porém não foram capazes de encontrar soluções para as maiores (e mais densas) instâncias em um intervalo de tempo de 1 hora de processamento.

A Tabela 5.10 apresenta as soluções inteiras (PMR_{PLI}^{LCouDC}) obtidas para as instâncias das classes B e F. Nessa tabela, a coluna *Perc* apresenta o percentual da solução obtida em relação a melhor solução conhecida ($(v(PMR_{PLI}^{LCouDC})/v(Opt))*100$). Os resultados obtidos pelas abordagens baseadas na GC são comparados aos resultados apresentados pelas diferentes heurísticas apresentadas em Glover et al. (2002). Como pode ser observado, as GCs, com a LagClus e a DecLag, apresentaram melhores resultados para todas as instâncias da classe B. Já para as instâncias da classe F, a

GC com a LagClus apresentou resultados para as instâncias com densidade de até 50%, e a GC com a DecLag apenas para a instância com 10% de densidade. Porém, em ambos os casos as soluções apresentados pelas GCs propostas foram melhores. O tempo médio para resolução das instâncias da classe B foi de 1,84 e 96,42 seg. para a GC com a LagClus e a DecLag, respectivamente. Para as instâncias da classe F, o tempo foi de aproximadamente 75 minutos para ambos os casos. Glover et al. (2002) não apresentam o tempo médio de processamento. O método descrito em Billionnet e Elloumi (2007) não foi capaz de resolver as instâncias da classe F, e segundo os autores, todas as soluções ótimas foram encontradas para as instâncias da classe B com um limite de tempo de 2 horas de processamento.

Tabela 5.10 - Soluções obtidas para as instâncias das classes B e F.

Inst.	v(Opt)	Glover et al. (2002) - Perc.					GC + LagClus		GC + DecLag	
		DDT	A2n	A2t	V3n	V3t	v(PMR _{PLI} ^C)	Perc.	v(PMR _{PLI} ^{DC})	Perc.
1b	133	73,7	100	78,9	10,5	10,5	133	100	133	100
2b	121	95,0	75,2	86,8	5,0	5,0	121	100	121	100
3b	118	47,5	86,4	80,5	0	0	118	100	118	100
4b	129	66,7	78,3	78,3	0	0	129	100	129	100
5b	150	70,0	100	60,0	0,7	0,7	150	100	150	100
6b	146	43,2	77,4	72,6	2,7	2,7	133	91,1	128	87,7
7b	160	56,3	100	100	0,6	0,6	160	100	160	100
8b	145	61,4	80,7	80,7	0	0	145	100	145	100
9b	137	70,1	92,7	75,9	0	0	135	98,5	133	97,1
10b	154	65,6	78,6	78,6	0	0	147	95,5	147	95,5
1f	61194	99,2	77,9	76,9	92,9	92,8	60959	99,6	60959	99,6
2f	100161	99,3	80,4	80,8	92,7	90,4	99836	99,7	-	-
3f	138035	98,7	78,6	79,7	92,7	92,1	137716	99,8	-	-
4f	172771	98,7	82,0	83,9	94,4	94,4	-	-	-	-
5f	190507	98,9	87,8	77,5	95,4	94,2	-	-	-	-

A Tabela 5.11 apresenta as soluções obtidas para as instâncias da classe D. Os *gaps* obtidos pelas GCs propostas são comparados aos apresentados por Billionnet e Elloumi (2007). Apenas para a instância 1d o *gap* apresentado pela GC com a LagClus foi diferente do apresentado pela GC com a DecLag, porém ambos melhores do que os apresentados por Billionnet e Elloumi (2007), assim como para as demais instâncias. O tempo médio para resolução de cada instâncias foi de 1 hora e 75 minutos para a GC com a LagClus e a DecLag, respectivamente. O método apresentado por Billionnet e Elloumi (2007) utilizou um tempo médio de 10 minutos por instância. Nessa tabela, os *gaps* são calculados de acordo com a Equação 5.153. Glover et al. (2002) não apresentam resultados para essas instâncias.

Tabela 5.11 - Soluções obtidas para as instâncias da classe D.

Inst.	v(Opt)	Billionnet e Elloumi (2007)	GC + LagClus		GC + DecLag	
		Gap	v(PMR _{PLI} ^{LC})	Gap	v(PMR _{PLI} ^{DC})	Gap
1d	6333	4,1	6327	0,09	6299	0,54
2d	6579	10	6579	0	6579	0
3d	9261	7,6	9261	0	9261	0
4d	10727	8,1	10727	0	10727	0
5d	11626	8,7	11604	0,19	11604	0,19
6d	14207	7,2	14157	0,35	14157	0,35
7d	14476	8,3	14476	0	14476	0
8d	16352	6,1	16352	0	16352	0
9d	15656	8,7	15647	0,06	15647	0,06
10d	19102	6,9	19102	0	19102	0

Tabela 5.12 - Soluções obtidas para as instâncias das classes A, C e E.

Inst.	v(Opt)	GC + LagClus		GC + DecLag	
		v(PMR _{PLI} ^{LC})	Gap	v(PMR _{PLI} ^{DC})	Gap
1a	3414	3414	0	3414	0
2a	6063	6063	0	6063	0
3a	6037	6035	0,03	6035	0,03
4a	8598	8582	0,19	8483	1,34
5a	5737	5737	0	5737	0
6a	3980	3980	0	3980	0
7a	4541	4541	0	4541	0
8a	11109	11109	0	11109	0
1c	5058	5058	0	5058	0
2c	6213	6213	0	6213	0
3c	6665	6665	0	6665	0
4c	7398	7398	0	7398	0
5c	7362	7362	0	7362	0
6c	5824	5824	0	5824	0
7c	7225	7225	0	7211	0,19
1e	16464	16447	0,10	16447	0,10
2e	23395	23376	0,08	23376	0,08
3e	25243	25228	0,06	25228	0,06
4e	35594	35594	0	35594	0
5e	35154	35149	0,01	35149	0,01
Gaps médios (%)		0,02		0,09	

Os resultados obtidos para as instâncias das classes A, C e E são apresentados na Tabela 5.12. Para as instâncias das classes A e C, o método proposto por Billionnet e Elloumi (2007) obteve as soluções ótimas em um tempo limite de 2 horas por instância, porém não foi capaz de resolver as instâncias da classe E.

Já as GCs propostas, com a LagClus e DecLag, apresentaram praticamente os mesmos resultados, próximos ou iguais ao ótimo. O tempo médio de processamento apresentado pela GC com a LagClus para as instâncias das classes A, C e E foi de 1 minuto, 40 minutos e 1 hora, respectivamente. Já para a GC com a DecLag, o tempo médio de processamento para essas instâncias foi de 7 segundos, 4 minutos e 1 minuto, respectivamente. Nessa tabela, os *gaps* são calculados de acordo com a Equação 5.153. Glover et al. (2002) não apresentam resultados para essas instâncias.

5.10 Considerações finais

Neste capítulo foram apresentadas novas estratégias para representação e obtenção de limitantes e soluções para o problema de programação quadrática binária irrestrita. Foram realizados vários experimentos com instâncias de diferentes características cuja obtenção de soluções apresenta um alto grau de dificuldade.

A relaxação linear do modelo PQL com a restrição de corte (PQLC) apresentou bons resultados para instâncias maiores com alta densidade, porém o CPLEX não foi capaz de encontrar soluções em alguns casos. Já as relaxações lagrangianas tradicionais, baseadas nas arestas positivas, apresentaram bons resultados para as instâncias com 500 variáveis, e resultados medianos para as demais. As demais relaxações apresentaram resultados ruins em todos os casos.

As abordagens baseadas na Relaxação Lagrangiana com *Clusters* (LagClus) apresentaram bons resultados para as instâncias com até 200 variáveis, principalmente para as instâncias com baixa densidade. Já para as instâncias com 500 variáveis, tais métodos falharam, principalmente pela dificuldade encontrada pelo CPLEX em resolver os subproblemas dentro de um tempo aceitável.

Novas alternativas baseadas na técnica de Decomposição Lagrangiana (DecLag) também foram apresentadas neste capítulo. Nesses casos, excelentes resultados foram obtidos para as instâncias com até 200 variáveis, com várias densidades, e para a instância com 500 variáveis com baixa densidade. Já para as instâncias com 500 variáveis e alta densidade, tais métodos não apresentaram soluções, pois o CPLEX

não foi capaz de resolver os subproblemas dentro de um tempo aceitável. Essas decomposições tratam o PQ por meio de modelos lineares representados por grafos, e copiam apenas as variáveis de decisão necessárias em cada subproblema. Além disso, a divisão do problema é realizada pelas heurísticas disponíveis na biblioteca METIS (KARYPIS; KUMAR, 1998), e os subproblemas são resolvidos pelo CPLEX (ILOG, 2006). Essas características permitem a resolução de problemas com até 500 variáveis e com várias densidades, e diferenciam as decomposições propostas das já existentes na literatura para problemas quadráticos (BILLIONNET et al., 1999; CHARDAIRE; SUTTER, 1995), que utilizam modelos pseudo-lineares, não utilizam a representação em grafos, a METIS e o CPLEX, e são limitadas a problemas com até 100 variáveis.

As abordagens baseadas na DecLag não foram comparados diretamente com a decomposição proposta por Chardaire e Sutter (1995) devido à indisponibilidade das instâncias utilizadas por esses autores. Entretanto, os resultados apresentam indícios de que as decomposições propostas são superiores, pois os modelos propostos com a representação em grafos e com o auxílio de uma heurística para particioná-los gerando um pequeno número de arestas inter-clusters, resulta em uma relaxação mais forte para o problema original, e conseqüentemente uma maior independência entre os subproblemas, o que permite a resolução de problemas maiores (com densidades variadas) com *gaps* de boa qualidade.

As melhores abordagens propostas para a LagClus e DecLag foram utilizadas em conjunto com a técnica de Geração de Colunas (GC), por meio da decomposição Dantzig-Wolfe (ver Seção 2.3). Dessa forma, além de apresentar duas alternativas para obtenção de limitantes, essas abordagens ainda apresentam soluções viáveis para o PQ. As soluções obtidas foram comparadas diretamente com as obtidas por outros métodos propostos recentemente na literatura, apresentando melhoras significativas para praticamente todas as instâncias utilizadas. Além disso, acredita-se que uma técnica mais eficiente para resolução dos subproblemas geradores de colunas possa resultar em melhores soluções para as instâncias tratadas neste trabalho e possivelmente para instâncias ainda maiores.

A abordagem baseada na posiforme foi apenas um início de uma busca por modelos que simplifiquem (e facilitem) o particionamento do grafo do PQ. Entretanto, maiores esforços ainda devem ser realizados para explorar essa técnica, pois os resultados apresentados, apesar de não serem tão bons quanto aqueles apresentados

pelos demais métodos propostos, parecem ser uma boa alternativa para resolver o PQ.

Enfim, os resultados obtidos indicam que os métodos apresentados contribuem com a exploração do PQ, e apresentam novas alternativas de relaxação e decomposição para problemas representados por grafos.

Todos os experimentos apresentados neste capítulo foram realizados em um *PC* com processador *AMD Athlon™ 64 3500* de 2.2 GHz e 1GB de memória RAM. Toda a implementação foi desenvolvida na linguagem C++ com chamadas às bibliotecas do *software* CPLEX 10.0 e da METIS.

As soluções obtidas por Billionnet e Elloumi (2007) foram executadas em um PC com processador *Intel Pentium IV* de 1.6 GHz e 1GB de memória RAM. Glover et al. (2002) não informam a especificação da máquina utilizada para execução de suas heurísticas.

6 PROBLEMA DE ROTULAÇÃO CARTOGRÁFICA DE PONTOS

O termo “rotulação cartográfica” consiste no processo de inserção de texto de objetos cartográficos em documentos do tipo mapa, diagrama, gráfico, etc. Geralmente, os objetos cartográficos podem ser pontos (cidades, hospitais, escolas, etc.), linhas (rios, estradas, etc.) e áreas (países, oceanos, etc.), que determinam a localização de aspectos naturais ou artificiais da terra (FREEMAN, 1991).

O Problema da Rotulação Cartográfica de Pontos (PRCP) é classificado na literatura como *NP-Hard* (FORMANN; WAGNER, 1991; MARKS; SHIEBER, 1991), e segundo Yamamoto (1998), o processo de rotulação cartográfica manual leva por volta de 1 ano para ser concluído. Logo, percebe-se que a rotulação cartográfica é uma importante tarefa na cartografia, principalmente nos sistemas de informações geográficas (WOLFF, 1999).

Segundo Ribeiro (2007), o PRCP consiste em rotular os pontos de um mapa evitando as sobreposições (conflitos) dos rótulos. A Figura 6.1 apresenta um mapa no qual são exibidas algumas sedes municipais presentes no estado do Espírito Santo. As setas indicam algumas sobreposições de rótulos, o que inviabiliza a leitura.



Figura 6.1 - Exemplo de um mapa com rótulos sobrepostos.

Fonte: Adaptada de Ribeiro e Lorena (2008b).

Na Figura 6.1, pode-se observar (setas) alguns conflitos entre os rótulos. Esses conflitos podem ser eliminados pelo deslocamento dos rótulos, afastando-os uns dos outros. Entretanto, deve-se garantir que os rótulos estejam próximos dos respectivos pontos. Assim, surge um conceito amplamente explorado no PRCP, que é o de “posições candidatas”. As posições candidatas representam o conjunto de todas as possíveis posições que o rótulo de um determinado ponto pode ocupar, obedecendo a uma padronização cartográfica. Logo, percebe-se que esse conceito limita o deslocamento dos rótulos.

Como pode ser observado nos principais trabalhos referentes ao PRCP, a padronização cartográfica mais utilizada é a proposta por Christensen et al. (1995). Essa padronização (ver Figura 6.2) considera 8 possíveis posições candidatas para cada ponto, sendo dada a cada uma dessas posições uma preferência cartográfica. Na Figura 6.2, cada posição possui um número, sendo a posição 1 a de maior interesse para a cartografia, ou seja, quanto menor for o número, maior será a preferência cartográfica.

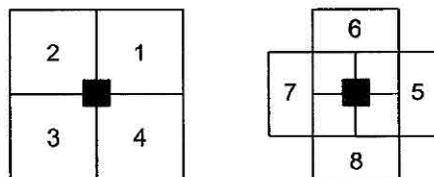


Figura 6.2 - Padronização cartográfica proposta por Christensen et al. (1995).

Fonte: Adaptada de Christensen et al. (1995).

A partir da padronização cartográfica, o PRCP pode ser definido como um problema de otimização combinatória que consiste em combinar as posições dos rótulos dos pontos em determinadas posições candidatas, de tal forma que atenda a um conjunto de restrições, minimizando ou maximizando uma função objetivo.

De uma forma geral, essas restrições são baseadas nas possíveis sobreposições de rótulos, e conseqüentemente podem ser modeladas sobre um “grafo de conflitos”. Segundo 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, ou seja, $V = x_i \in \{0,1\}$, $i = 1, \dots, |V|$, e as arestas representam suas relações lógicas tal que $x_i + x_j \leq 1$, $\forall (i, j) \in A$. Nesse caso, uma aresta no grafo de conflitos indica que suas variáveis binárias (os vértices referentes a cada aresta) não podem assumir o valor 1 simultaneamente.

No caso específico do PRCP, esse grafo pode ser definido como $G = (V, A)$, no qual $V = \{v_{1,1}, v_{1,2}, \dots, v_{1,P_1}, \dots, v_{N,1}, v_{N,2}, \dots, v_{N,P_N}\}$ representa o conjunto das posições candidatas (vértices), sendo N o número de pontos a serem rotulados e P_i ($i = 1, \dots, N$) o número de posições candidatas do ponto i , e $A = (v_{i,j}, v_{t,u}) : \{v_{i,j} \text{ e } v_{t,u} \in V, i \neq t \text{ ou } j \neq u\}$ o conjunto das arestas que indicam as sobreposições (conflitos). Logo, considerando um problema com 2 pontos e 4 posições candidatas para seus respectivos rótulos, pode-se gerar o grafo de conflitos assim como apresentado na Figura 6.3. Nessa figura, as arestas tracejadas apresentam os conflitos dos vértices $v_{1,3}$ com $v_{2,2}$ e $v_{1,4}$ com $v_{2,1}$ e $v_{2,2}$.

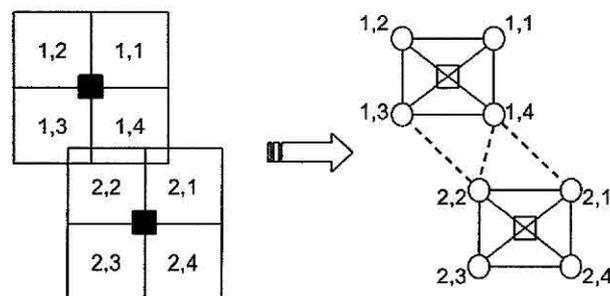


Figura 6.3 - Grafo de conflitos para o PRCP.

Fonte: Adaptada de Ribeiro (2007).

Os grafos de conflito estão diretamente relacionados ao Problema do Máximo Conjunto Independente de Vértices (PMCIIV). Esse problema tem como objetivo obter o maior subconjunto de vértices tal que todos eles sejam independentes entre si, ou seja, não apresentem nenhuma aresta de conflito. Logo, 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). Dessa forma, o PRCP pode ser formulado como um Problema do Máximo Conjunto Independente de Vértices - PMCIIV (STRIJK et al., 2000; ZORASTER, 1986; ZORASTER, 1990; ZORASTER, 1991).

Nessa abordagem, alguns pontos podem ficar sem seus respectivos rótulos, porém, sob o ponto de vista cartográfico, é interessante que todos os pontos sejam rotulados, mesmo que algumas sobreposições sejam inevitáveis (RIBEIRO, 2007). Além da abordagem baseada no PMCIIV, existem ainda 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), sendo que ambas rotulam todos os pontos.

O PMNRSC procura maximizar o número de pontos rotulados sem conflitos, sem considerar o aspecto visual do mapa, podendo apresentar áreas ilegíveis. Por outro lado, o PMNC procura rotular todos os pontos, minimizando os conflitos gerados. Essas abordagens são mais interessantes do ponto de vista cartográfico, pois garantem a rotulação de todos os pontos.

Na literatura, não são apresentadas formulações matemáticas baseadas na Programação Linear Inteira (PLI) para o PMNRSC. Já para o PMNC, existem 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 (2004).

6.1 Revisão da literatura

Considerando o PRCP como um PMCIV, destacam-se os trabalhos apresentados por Zoraster (1986), Zoraster (1990), Zoraster (1991) e Strijk et al. (2000). Esses trabalhos apresentam diferentes métodos para resolução do PRCP, além de algumas técnicas para redução do número de restrições. Já considerando o PRCP como um PMNRSC, pode-se destacar os mais recentes, e talvez mais importantes (sob o ponto de vista cartográfico) trabalhos referentes ao PRCP (ALVIM; TAILLARD, 2009; CHRISTENSEN et al., 1993; CHRISTENSEN et al., 1995; RIBEIRO, 2007; RIBEIRO; LORENA, 2008a; RIBEIRO; LORENA, 2008b; VERNER et al., 1997; YAMAMOTO et al., 2002; YAMAMOTO; LORENA, 2005). Como pode ser observado no texto a seguir, esses trabalhos apresentam uma acirrada disputa pelas melhores soluções para as instâncias propostas por Yamamoto et al. (2002).

Considerando o PMCIV, uma Relaxação Lagrangiana tradicional sobre o conjunto de restrições de adjacência foi apresentada em Zoraster (1990). Essa relaxação foi aplicada a instâncias pequenas produzindo bons resultados. Recentemente, Strijk et al. (2000) propuseram uma formulação baseada em programação inteira binária que utiliza restrições de corte para reduzir o número de restrições de conflito presentes nos modelos de Zoraster (1990) e Zoraster (1991). Esses cortes consistem em utilizar alguns conceitos de cliques, reduzindo assim o número total de restrições do problema original.

Strijk et al. (2000) ainda 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 vários métodos heurísticos baseados no *Simulated Annealing*, Busca em Vizinhança Diversificada, *k-Opt* e Busca Tabu, sendo essa última a que apresentou os melhores resultados.

Considerando agora o PRCP como um PMNRSC, Christensen et al. (1993) e Christensen et al. (1995) propuseram métodos de busca exaustiva que procuram por melhores soluções por meio de trocas das posições de rótulos previamente posicionados. Christensen et al. (1995) também apresentam um algoritmo guloso com sucessivas otimizações locais, e um algoritmo denominado *Discrete Gradient Descent*, que considera as posições alternativas dos rótulos. Como observado pelos autores, esse algoritmo encontra dificuldades para escapar de máximos locais.

Ainda considerando o PMNRSC, Verner et al. (1997) utilizaram um algoritmo genético para obtenção de soluções. Em seguida, Yamamoto et al. (2002) propuseram um eficiente algoritmo baseado na Busca Tabu, que forneceu excelentes resultados quando comparados com a literatura. Yamamoto e Lorena (2005) propuseram um algoritmo exato para o PMNRSC, e aplicaram também o Algoritmo Genético Construtivo (ver Subseção 2.6.1). O algoritmo exato apresentou resultados apenas para instâncias com até 25 pontos. Já o AGC foi aplicado a instâncias com até 1000 pontos, apresentando bons resultados quando comparados aos obtidos pela Busca Tabu proposta por Yamamoto et al. (2002).

Já a abordagem que considera o PRCP como um PMNC foi introduzida por Ribeiro e Lorena (2004). O PMNC considera a minimização do número de conflitos, e como observado pelos autores, em alguns casos as soluções encontradas são limitantes inferiores para o PMNRSC. Um GRASP (*Greedy Randomized Adaptive Search Procedure*) para resolução do PRCP foi apresentado em Cravo et al. (2008). Nesse método, os autores mesclam os conceitos do PMNC com os do PMNRSC. Os resultados obtidos superaram os apresentados em Yamamoto e Lorena (2005).

Ribeiro e Lorena (2008b) apresentam a Relaxação Lagrangiana com *Clusters - LagClus* (ver Subseção 2.8.2) para o PRCP. Nesse caso, os resultados obtidos superaram os encontrados até então na literatura. Na sequência, Ribeiro e Lorena (2008a) utilizam a geração de colunas em conjunto com a LagClus (ver Subseção 5.7.1), e os resultados obtidos são melhores que os apresentados pela LagClus de forma isolada.

Por fim, mais recentemente, a metaheurística POPMUSIC (*Partial Optimization Metaheuristic under Special Intensification Conditions*) proposta por Taillard e Voss (2001) foi utilizada por Alvim e Taillard (2009) para resolução das instâncias propostas por Yamamoto et al. (2002). De uma forma geral, pode-se dizer que, dada uma solução viável para o problema, a POPMUSIC otimiza partes dessa solução localmente. Esse processo é repetido até que um mínimo ou máximo local seja alcançado. Outros métodos apresentados na literatura são utilizados para obtenção das soluções viáveis iniciais, e vários tamanhos foram utilizados para definir as “partes” do problema original. Os resultados obtidos superam os apresentados pelos demais trabalhos encontrados na literatura até o presente momento.

6.2 Decomposição do PRCP em *clusters*

Como apresentado na Subseção 2.8.3, o primeiro passo para aplicação da DecLag é a determinação da forma de particionar o grafo do problema e copiar as variáveis necessárias.

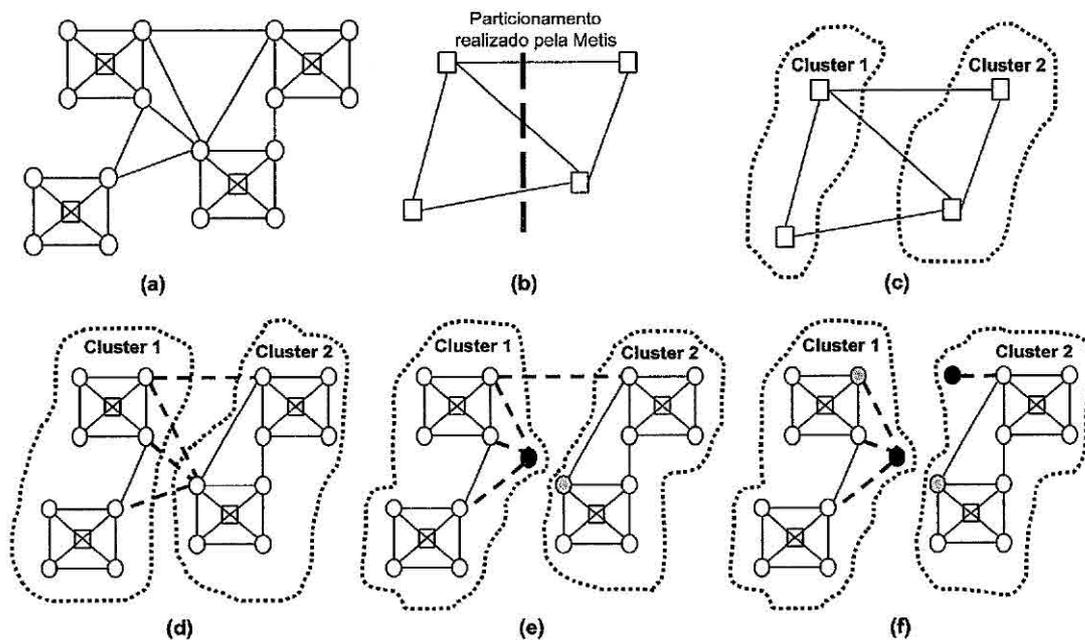


Figura 6.4 - Decomposição do PRCP em *clusters*.

Logo, considerando o grafo de conflitos G (Figura 6.4a), deve-se definir a melhor estratégia para particioná-lo. Sendo assim, uma estratégia interessante é a técnica de contração de vértices apresentada em Goldberg e Luna (2005), que consiste em agrupar todas as posições candidatas do ponto i , formando um único vértice (ver Figura 6.4b). Na Figura 6.4, os quadrados representam os pontos a serem rotulados, e os círculos as posições candidatas dos respectivos pontos.

Como observado por Ribeiro (2007), essa técnica gera um grafo de conflitos entre pontos e não entre posições candidatas. Logo, o particionamento do grafo de conflitos entre pontos irá gerar subproblemas (Figura 6.4c) nos quais os conflitos entre as posições candidatas de um mesmo ponto sejam preservados, resultando assim em uma relaxação mais forte para o PRCP.

Após o particionamento de G , as contrações são expandidas (Figura 6.4d), gerando novamente o grafo de conflitos original, surgindo porém as arestas *inter-clusters* (arestas tracejadas na Figura 6.4d) e os subproblemas (*clusters* 1 e 2).

Seguindo as idéias da DecLag, deve-se determinar agora quais serão os vértices copiados. Assim, diferentemente das abordagens apresentadas para o PQ (ver Capítulo 5), foi utilizada a heurística proposta por Sachdeva (2004). De uma forma geral, essa heurística consiste em copiar inicialmente o vértice que apresenta o maior número de arestas *inter-clusters*, e repetir esse processo até que todas as cópias necessárias sejam realizadas, ou seja, até que todas as arestas *inter-clusters* sejam eliminadas.

Como pode ser observado na Figura 6.4e, o primeiro vértice copiado apresenta 3 arestas *inter-clusters* (o vértice preto é uma cópia do vértice cinza). A simples cópia do vértice cinza para o *cluster* 1 (vértice preto - Figura 6.4e) já elimina três arestas *inter-clusters*. Porém, ainda falta uma, então o processo é repetido, e nesse caso, tem-se a Figura 6.4f.

Pode-se perceber que agora são apresentados dois *clusters* independentes, e caso seja garantido que as variáveis referentes aos vértices copiados sejam iguais às originais, ou seja, os vértices pretos (cópias) sejam iguais aos respectivos cinzas (originais), o problema original estará decomposto em 2 *clusters*.

6.3 DecLag para o PRCP como um PMCIV

Inicialmente, optou-se por aplicar a DecLag à formulação do PRCP baseada no PMCIV devido ao grande número de aplicações desse problema. Sendo assim, é tratado um modelo baseado no apresentado por Strijk et al. (2000), que é uma formulação comum para o PMCIV.

PMCIV: $v(\text{PMCIV}) = \text{Maximizar:}$

$$\sum_{i=1}^m \sum_{j=1}^{P_i} \omega_{i,j} x_{i,j} \quad (6.1)$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} \leq 1 \quad \forall i = 1, \dots, N \quad (6.2)$$

$$x_{i,j} + x_{t,u} \leq 1 \quad \forall i = 1, \dots, N; \forall j = 1, \dots, P_i; i \neq t; (t, u) \in S_{i,j} \quad (6.3)$$

$$x_{i,j}, x_{t,u} \in \{0, 1\} \quad \forall i = 1, \dots, N; \forall j = 1, \dots, P_i; (t, u) \in S_{i,j} \quad (6.4)$$

Nesse modelo, para cada ponto i existe um número fixo P_i de posições candidatas. Cada posição candidata é representada por uma variável binária $x_{i,j}$ onde $i = 1, \dots, m$, sendo m o número de pontos a serem rotulados, e $j = 1, \dots, P_i$. Se $x_{i,j} = 1$, a posição candidata j do ponto i será alocada (receberá o rótulo do ponto i), caso contrário, $x_{i,j} = 0$. $\omega_{i,j}$ indica a preferência cartográfica da posição candidata j do ponto i , ou seja, é uma penalidade.

Cada posição candidata $x_{i,j}$ possui um conjunto $S_{i,j}$ de pares de índices (t, u) de posições candidatas $x_{t,u}$ conflitantes com $x_{i,j}$. Para cada ponto i , apenas uma posição candidata deve ser selecionada, ou seja, no máximo uma das posições candidatas receberá o valor 1 (restrição 6.2). A restrição 6.3 garante que duas posições candidatas conflitantes não sejam selecionadas simultaneamente, e a restrição 6.4 garante que todas as variáveis de decisão sejam binárias. Vale destacar que a restrição 6.2 poderia ser tratada “dentro” da restrição 6.3 (retirando as condições $i \neq j$). Entretanto, essa restrição é descrita explicitamente para facilitar a adaptação do modelo ao PRCP.

Como apresentado na Figura 6.4, o grafo G pode ser decomposto em n ($n \leq m$) *clusters* independentes, e para isso, tem-se $V = V_1 \cup V_2 \cup \dots \cup V_n$, onde $V_i \cap V_j = \emptyset$ ($\forall i, j = 1, \dots, n; i \neq j$). $G_k = (V_k, A_k)$, $k = 1, \dots, n$, e $X_k = V - V_k$, $k = 1, \dots, n$. Além disso, cada *cluster* k ainda possui um conjunto de vértices copiados C_k , formado pela aplicação da heurística de Sachdeva (2004). Logo, o PMCIV pode ser reescrito como sendo:

$PMCIV^n$: $v(PMCIV^n) = \text{Maximizar:}$

$$\sum_{k=1}^n \left(\sum_{(i,j) \in V_k} \omega_{i,j} x_{i,j} \right) \quad (6.5)$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} \leq 1 \quad \forall i \in V_k; k = 1, \dots, n \quad (6.6)$$

$$x_{i,j} + x_{t,u} \leq 1 \quad \forall (i,j) \in V_k; i \neq t; (t,u) \in V_k; (t,u) \in S_{i,j}; k = 1, \dots, n \quad (6.7)$$

$$x_{i,j} + x_{t,u}^k \leq 1 \quad \forall (i,j) \in V_k; (t,u) \in C_k; (t,u) \in S_{i,j}; k = 1, \dots, n \quad (6.8)$$

$$x_{t,u} = x_{t,u}^k \quad \forall (t,u) \in C_k; (t,u) \in X_k; k = 1, \dots, n \quad (6.9)$$

$$x_{i,j}, x_{t,u}, x_{t,u}^k \in \{0, 1\} \quad \forall (i,j) \in V_k; (t,u) \in V_k \cup C_k; k = 1, \dots, n \quad (6.10)$$

Nesse modelo, as variáveis $x_{t,u}^k$ representam as cópias do vértice (t,u) no *cluster* k . Note que as cópias não são consideradas na função objetivo, ou seja, seu coeficiente é zero. As restrições 6.6 e 6.7 tratam apenas as arestas cujos vértices são internos ao *cluster* (sub-grafo) k (arestas *intra-cluster*). Já a restrição 6.8 trata as arestas cujos vértices estão em *clusters* distintos (arestas *inter-clusters*), que no caso da decomposição passam a ligar os vértices copiados (arestas tracejadas na Figura 6.4f).

A restrição 6.9 é uma restrição de cópia que garante a igualdade entre as variáveis originais e suas cópias. Logo, sua relaxação no sentido lagrangiano, por meio do vetor de multiplicadores $\vec{\alpha}$ ($\vec{\alpha}$ irrestrito), permite que o problema $PMCIV^n$, (indiretamente o PMCIV) possa ser dividido em n subproblemas independentes. Assim, para $k = 1, \dots, n$, tem-se:

$DC_\alpha PMCIV_k$: $DC_\alpha PMCIV_k = \text{Maximizar:}$

$$\sum_{(i,j) \in V_k} \left(\omega_{i,j} - \sum_{d \neq k} \alpha_{i,j}^d \right) x_{i,j} + \sum_{(t,u) \in C_k} (\alpha_{t,u}^d) x_{t,u}^k \quad (6.11)$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} \leq 1 \quad \forall i \in V_k \quad (6.12)$$

$$x_{i,j} + x_{t,u} \leq 1 \quad \forall (i,j) \in V_k; i \neq t; (t,u) \in V_k; (t,u) \in S_{i,j} \quad (6.13)$$

$$x_{i,j} + x_{t,u}^k \leq 1 \quad \forall (i,j) \in V_k; (t,u) \in C_k; (t,u) \in S_{i,j} \quad (6.14)$$

$$x_{i,j}, x_{t,u}, x_{t,u}^k \in \{0, 1\} \quad \forall (i,j) \in V_k; (t,u) \in V_k \cup C_k \quad (6.15)$$

Por fim, a solução da relaxação do problema PMCIV em n subproblemas é dada pela Equação 6.16, e o seu dual lagrangiano é apresentado na Equação 6.17.

$$DC_\alpha PMCIV^n : \quad v(DC_\alpha PMCIV^n) = \sum_{k=1}^n v(DC_\alpha PMCIV_k) \quad (6.16)$$

$$DDC_\alpha PMCIV^n : v(DDC_\alpha PMCIV^n) = \text{Min}\{v(DC_\alpha PMCIV^n)\} \\ \alpha \text{ irrestrito} \quad (6.17)$$

Assim como apresentado na Seção 5.6 e na Subseção 5.8.2, esse problema pode ser resolvido pelo algoritmo de subgradientes (ver Seção 2.9). Nesse caso, ainda foi utilizada uma heurística lagrangiana para obter soluções viáveis para o PMCIV a cada iteração do algoritmo de subgradientes, ou seja, diferentemente das decomposições apresentadas para o PQ (ver Capítulo 5), que apresentam limitantes superiores para o problema, para o PMCIV é utilizada essa heurística para calcular, a cada iteração do algoritmo, o limitante inferior (solução viável, nesse caso). Essa heurística é apresentada na Figura 6.5.

Na primeira parte dessa heurística (linhas 6 e 7), caso os valores das variáveis x na solução relaxada (nos *clusters*) sejam iguais a zero, o valor da variável na solução primal também será zero. Caso contrário (linhas 8 a 10), se todas as cópias tiverem o mesmo valor que a original, o valor na solução primal será o valor da original na

solução relaxada, e se alguma cópia possuir um valor diferente, será zero (linhas 11 a 13). Por fim, é feito um refinamento para cada ponto não rotulado (17 a 23). Para cada posição candidata não selecionada é verificado se o seu valor pode ser modificado de zero para um, ou seja, verifica se os pontos não rotulados poderão ser rotulados sem que ocorram conflitos.

```

1.  $x_{i,j}^*$ : valor de  $x_{i,j}$  na solução inteira (viável) do problema;
2.  $x_{i,j}$ : valor de  $x_{i,j}$  no subproblema do vértice  $(i,j)$ ;
3.  $x_{i,j}^k$ : valor da cópia de  $x_{i,j}$  no subproblema  $k$ ;
4. PARA  $i \leftarrow 1$  ATÉ  $N$  FAÇA
5.   PARA  $j \leftarrow 1$  ATÉ  $P_i$  FAÇA
6.     SE  $(x_{i,j} = 0)$  ENTÃO
7.        $x_{i,j}^* \leftarrow 0$ ;
8.     SENÃO
9.       SE  $(x_{i,j} = x_{i,j}^k, \forall k \leftarrow 1, \dots, n)$  ENTÃO
10.         $x_{i,j}^* \leftarrow x_{i,j}$ ;
11.      SENÃO
12.         $x_{i,j}^* \leftarrow 0$ ;
13.      FIM-SE;
14.    FIM-PARA;
15.  FIM-PARA;
16. FIM-PARA;
17. PARA  $i \leftarrow 1$  ATÉ  $N$  FAÇA
18.   PARA  $j \leftarrow 1$  ATÉ  $P_i$  FAÇA
19.     SE  $(x_{i,j}^* = 0) \text{ E } (x_{t,u}^* = 0, \forall x_{t,u} \in S_{i,j})$  ENTÃO
20.        $x_{i,j}^* = 1$ ;
21.     FIM-SE;
22.   FIM-PARA;
23. FIM-PARA;

```

Figura 6.5 - Heurística lagrangiana para o PMCIV.

6.4 DecLag para o PRCP como um PMNRSC

Inicialmente, para aplicação da Decomposição Lagrangiana - DecLag (ver Subseção 2.8.3) para resolução do Problema do Máximo Número de Rótulos Sem Conflitos (PMNRSC), deve-se utilizar uma formulação matemática. Entretanto, como citado anteriormente, na literatura não são apresentadas formulações baseadas na PLI para o PMNRSC.

Sendo assim, pode-se considerar o conceito de cobertura proveniente do problema das p -Medianas (GOLDBARG; LUNA, 2005; WOLSEY, 1998), ou seja, “o ponto i está coberto?”. Dessa forma, pode-se representar o PMNRSC de acordo com a seguinte formulação:

PMNRSC: $PMNRSC = \text{Maximizar:}$

$$\sum_{i=1}^m \sum_{j=1}^{P_i} \omega_{i,j} x_{i,j} - \sum_{i=1}^m z_i \quad (6.18)$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \quad \forall i = 1, \dots, N \quad (6.19)$$

$$x_{i,j} + x_{t,u} - z_i \leq 1 \quad \forall i = 1, \dots, N; \forall j = 1, \dots, P_i; i \neq t; (t, u) \in S_{i,j} \quad (6.20)$$

$$x_{i,j}, x_{t,u}, z_i \in \{0, 1\} \quad \forall i = 1, \dots, N; \forall j = 1, \dots, P_i; (t, u) \in S_{i,j} \quad (6.21)$$

Nesse modelo, as variáveis z_i (para cada ponto $i = 1, \dots, m$) indicam se o rótulo do ponto i está sobreposto a outro qualquer, ou seja, $z_i = 1$ caso o rótulo do ponto i esteja sobreposto, e 0 caso contrário. A atribuição do valor 0 ou 1 às variáveis z_i é garantida pela restrição 6.20.

É interessante destacar que as variáveis de sobreposição z_i aparecem subtraindo na função objetivo, e esta ao ser maximizada deverá procurar por soluções cujo valor dessas variáveis sejam iguais a zero. Essa formulação é semelhante às propostas por Zoraster (1990) e Ribeiro (2007), porém ela permite posicionar todos os rótulos maximizando o número de rótulos sem conflitos.

Considerando o grafo de conflitos apresentado na Figura 6.6a, tem-se as respectivas restrições 6.20 na Figura 6.6b.

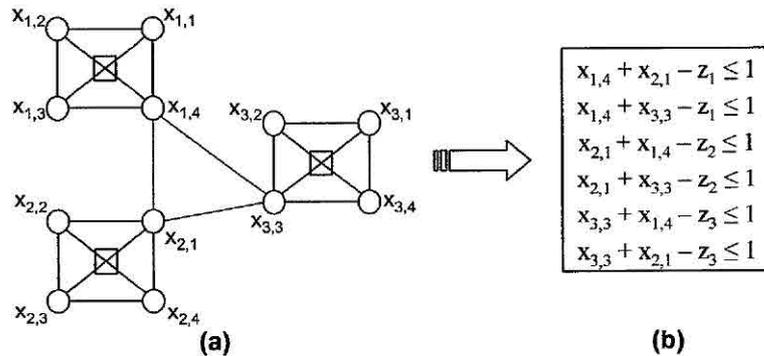


Figura 6.6 - Exemplo de grafo e restrições para o PMNRSC.

Como pode ser observado na Figura 6.6, caso $x_{1,4} = x_{2,1} = 1$, o ponto 1, representado por z_1 , será sobreposto (pelo ponto 2). Conseqüentemente, $x_{2,1} = x_{1,4} = 1$, e o ponto 2, representado por z_2 , também será sobreposto (pelo ponto 1). As restrições 6.20 apresentadas na Figura 6.6b garantem então que se houver sobreposição as variáveis z receberão os valores corretamente. Note que, como a função objetivo visa maximizar o número de rótulos livres, porém rotulando todos os pontos, os somatórios das variáveis x_{ij} (ver 6.18) sempre será igual ao número total de pontos (garantido pela restrição 6.19), ou seja, nesse exemplo, igual a 3, e o número de sobreposições será dado pelo somatório das variáveis z . Logo, o valor da função objetivo será exatamente o número de rótulos sem conflitos. No exemplo da Figura 6.6, a função objetivo será: Maximizar $3 - z_1 - z_2 - z_3$. Dessa forma, caso $x_{1,4} = x_{2,1} = x_{3,3} = 1$, a restrição 6.20 irá garantir que $z_1 = z_2 = z_3 = 1$, ou seja, todos os pontos foram rotulados, porém todos estão sobrepostos, e conseqüentemente não existem rótulos livres, e a FO será igual a zero.

Considerando então a definição de variáveis e a decomposição apresentada na seção anterior, o PMNRSC pode ser reescrito como sendo:

$PMNRSC^n$: $v(PMNRSC^n) = \text{Maximizar:}$

$$\sum_{k=1}^n \left(\sum_{(i,j) \in V_k} \omega_{i,j} x_{i,j} - \sum_{\substack{(i,j) \in V_k; \\ j=1, \dots, P_i}} z_i \right) \quad (6.22)$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \quad \forall i \in V_k; k = 1, \dots, n \quad (6.23)$$

$$x_{i,j} + x_{t,u} - z_i \leq 1 \quad \forall (i,j) \in V_k; i \neq t; (t,u) \in V_k; (t,u) \in S_{i,j}; k = 1, \dots, n \quad (6.24)$$

$$x_{i,j} + x_{t,u}^k - z_i \leq 1 \quad \forall (i,j) \in V_k; (t,u) \in C_k; (t,u) \in S_{i,j}; k = 1, \dots, n \quad (6.25)$$

$$x_{i,j} + x_{t,u}^k - z_t^k \leq 1 \quad \forall (i,j) \in V_k; (t,u) \in C_k; (t,u) \in S_{i,j}; k = 1, \dots, n \quad (6.26)$$

$$x_{t,u} = x_{t,u}^k \quad \forall (t,u) \in C_k; (t,u) \in X_k; k = 1, \dots, n \quad (6.27)$$

$$z_t = z_t^k \quad \forall (t,u) \in C_k; (t,u) \in X_k; k = 1, \dots, n \quad (6.28)$$

$$x_{i,j}, x_{t,u}, x_{t,u}^k, z_t^k \in \{0, 1\} \quad \forall (i,j) \in V_k; (t,u) \in V_k \cup C_k; k = 1, \dots, n \quad (6.29)$$

Nesse modelo, as variáveis $x_{t,u}^k$ e z_t^k representam as cópias do vértice (t,u) e da variável z_t no *cluster* k , respectivamente. De forma análoga à descrita na seção anterior, as cópias não são consideradas na função objetivo, ou seja, seu coeficiente é zero. As restrições 6.23 e 6.24 tratam apenas as arestas cujos vértices são internos ao *cluster* (sub-grafo) k (arestas *intra-cluster*). Já as restrições 6.25 e 6.26 tratam as arestas cujos vértices estão em *clusters* distintos (arestas *inter-clusters*). As restrições 6.27 e 6.28 são as restrições de cópia.

Relaxando as restrições 6.27 e 6.28 no sentido lagrangiano por meio dos vetores de multiplicadores $\vec{\alpha}$ e $\vec{\beta}$ ($\vec{\alpha}$ e $\vec{\beta}$ irrestritos), o problema $PMNRSC^n$, (indiretamente o PMNRSC) pode ser dividido em n subproblemas independentes. Logo, para $k = 1, \dots, n$, tem-se:

$DC_{\alpha\beta}PMNRSC_k$: $v(DC_{\alpha\beta}PMNRSC_k) = \text{Maximizar:}$

$$\begin{aligned} & \sum_{(i,j) \in V_k} \left(\omega_{i,j} - \sum_{d \neq k} \alpha_{i,j}^d \right) x_{i,j} + \sum_{(t,u) \in C_k} (\alpha_{t,u}^d) x_{t,u}^k \\ & - \sum_{(i,j) \in V_k} \left(z_i - \sum_{d \neq k} \beta_i^d \right) z_i - \sum_{(t,u) \in C_k} (\beta_t^d) z_t^k \end{aligned} \quad (6.30)$$

Sujeito a:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \quad \forall i \in V_k \quad (6.31)$$

$$x_{i,j} + x_{t,u} - z_i \leq 1 \quad \forall (i,j) \in V_k; (t,u) \in V_k; i \neq t; (t,u) \in S_{i,j} \quad (6.32)$$

$$x_{i,j} + x_{t,u}^k - z_i \leq 1 \quad \forall (i,j) \in V_k; (t,u) \in C_k; (t,u) \in S_{i,j} \quad (6.33)$$

$$x_{i,j} + x_{t,u}^k - z_t^k \leq 1 \quad \forall (i,j) \in V_k; (t,u) \in C_k; (t,u) \in S_{i,j} \quad (6.34)$$

$$x_{i,j}, x_{t,u}, x_{t,u}^k, z_t^k \in \{0, 1\} \quad \forall (i,j) \in V_k; (t,u) \in V_k \cup C_k \quad (6.35)$$

Por fim, a solução da relaxação do problema PMNRSC em n subproblemas é dada pela Equação 6.36, e o seu dual lagrangiano é apresentado na Equação 6.37.

$$DC_{\alpha\beta}PMNRSC^n : \quad v(DC_{\alpha\beta}PMNRSC^n) = \sum_{k=1}^n v(DC_{\alpha\beta}PMNRSC_k) \quad (6.36)$$

$$DDC_{\alpha\beta}PMNRSC^n : v(DDC_{\alpha\beta}PMNRSC^n) = \text{Min}\{v(DC_{\alpha\beta}PMNRSC^n)\}, \alpha, \beta \text{ irrestritos} \quad (6.37)$$

Assim como apresentado na seção anterior, esse problema também pode ser resolvido pelo algoritmo de subgradientes. Nesse caso, uma heurística lagrangiana também é utilizada para obtenção dos limitantes inferiores. Essa heurística é apresentada na Figura 6.7.

```

1.  $x_{i,j}^*$ : valor de  $x_{i,j}$  na solução inteira (viável) do problema;
2.  $x_{i,j}$ : valor de  $x_{i,j}$  no subproblema do vértice  $(i,j)$ ;
3.  $p$ : posição candidata selecionada;
4.  $r$ : posição candidata selecionada antes da melhor troca;
5.  $q$ : posição candidata selecionada depois da melhor troca;
6. PARA  $i \leftarrow 1$  ATÉ  $N$  FAÇA
7.   PARA  $j \leftarrow 1$  ATÉ  $P_i$  FAÇA
8.      $x_{i,j}^* \leftarrow x_{i,j}$ 
9.   FIM-PARA;
10. FIM-PARA;
11.  $fo^* \leftarrow$  CALCULAR (a função objetivo);
12. FAÇA
13.    $melhorou \leftarrow$  falso;
14.   PARA  $i \leftarrow 1$  ATÉ  $N$  FAÇA
15.     PARA  $j \leftarrow 1$  ATÉ  $P_i$  FAÇA
16.       SE  $(x_{i,j}^* = 1)$  ENTÃO
17.          $p \leftarrow j$ ;
18.       FIM-SE;
19.     FIM-PARA;
20.     PARA  $j \leftarrow 1$  ATÉ  $P_i$  FAÇA
21.       SE  $(j \neq p)$  ENTÃO
22.          $x_{i,p} \leftarrow 0$ ;
23.          $x_{i,j} \leftarrow 1$ ;
24.          $fo =$  CALCULAR (a função objetivo);
25.         SE  $(fo > fo^*)$  ENTÃO
26.            $r \leftarrow p$ ;
27.            $q \leftarrow j$ ;
28.            $fo^* \leftarrow fo$ ;
29.            $melhorou \leftarrow$  verdadeiro;
30.         FIM-SE;
31.        $x_{i,p} \leftarrow 1$ ;
32.        $x_{i,j} \leftarrow 0$ ;
33.     FIM-PARA;
34.   FIM-PARA;
35.   SE ( $melhorou$ ) ENTÃO
36.      $x_{i,r}^* \leftarrow 0$ ;
37.      $x_{i,q}^* \leftarrow 1$ ;
38.   FIM-SE;
39. ENQUANTO ( $melhorou$ );

```

Figura 6.7 - Heurística lagrangiana para o PMNRSC.

Essa é uma heurística gulosa que consiste basicamente em verificar a melhor troca de posições candidatas enquanto a solução do problema melhorar. Na primeira parte dessa heurística (linhas 6 a 10) é montada a solução obtida pela relaxação. Nas

linhas 15 a 19 é verificada a posição candidata selecionada para o ponto i . Nas linhas 20 a 33 essa posição selecionada é trocada por todas as outras possíveis, e caso uma melhor solução seja obtida, as posições modificadas são armazenadas, e caso contrário, a modificação é desfeita (linhas 31-32). Esse processo é repetido para todos os pontos (linhas 14-34). Enquanto for encontrada uma melhor solução, todo o processo é repetido (linhas 12 a 39).

6.5 Experimentos computacionais

Para avaliar o desempenho das decomposições propostas neste capítulo, foram utilizadas as instâncias (com 100, 250, 500, 750 e 1000 pontos) propostas por Yamamoto et al. (2002), e as instâncias com 13206 pontos propostas por Alvim e Taillard (2009). Essas instâncias são consideradas nos principais trabalhos referentes ao PRCP (ver Seção 6.1), e estão disponíveis em <http://www.lac.inpe.br/~lorena/instancias.html> e http://mistic.heig-vd.ch/taillard/problemes.dir/label/swiss_map_label.zip.

Como considerado em Zoraster (1990), Christensen et al. (1995), Yamamoto e Lorena (2005), Ribeiro (2007), Ribeiro e Lorena (2008a), Ribeiro e Lorena (2008b) e Alvim e Taillard (2009), entre outros, em todos os problemas não foram consideradas preferências cartográficas, ou seja, considerou-se o custo ou penalidade igual a 1 ($\omega_{i,j} = 1$) para todas as posições candidatas, sendo o número total dessas posições igual a 4.

Assim como no Capítulo 5, a divisão dos grafos foi realizada pelas heurísticas disponíveis na biblioteca METIS (KARYPIS; KUMAR, 1998). Todos os resultados apresentados a seguir foram obtidos a partir do grafo resultante da redução proposta por Wagner et al. (2001), assim como apresentado por Ribeiro (2007), Ribeiro e Lorena (2008a), Ribeiro e Lorena (2008b) e Alvim e Taillard (2009).

A Tabela 6.1 apresenta os resultados obtidos pelo modelo PMCIV para as instâncias com 13206 pontos. Para as demais instâncias (100, 250, 500, 750 e 1000 pontos) todas as soluções ótimas foram obtidas em um tempo computacional médio inferior a 1 segundo, tanto pelo CPLEX quanto pela $DC_{\alpha}PMCIV^n$. Nas Tabelas 6.2, 6.3 e 6.4 são apresentados os resultados obtidos para o PMNRSC com as instâncias de 500, 750 e 1000 pontos, respectivamente.

Tabela 6.1 - Soluções obtidas pelo PMCV para as instâncias com 13206 pontos.

Inst.	DC _α PMCV ⁿ										CPLEX		
	n	LB	UB	Gap	M. & P.	Subp.	Subg.	Total	LB	UB	Gap	Tempo	
1	20	12965*	12965	0	12,53	0,69	2,56	15,78	12965	12976,26	0,09	7810,53	
2	20	12958*	12958	0	8,50	0,34	1,66	10,50	12958	12959,33	0,01	7682,14	
3	20	12640*	12640	0	14,62	0,59	3,94	19,15	12637	12666,07	0,23	7873,06	
4	20	12439*	12439	0	18,19	0,73	6,09	25,01	12439	12477,93	0,31	8574,38	
5	20	13102*	13102	0	7,16	0,25	1,34	8,75	13102	13103,83	0,01	8036,94	
6	20	12550*	12550	0	16,73	0,67	7,72	25,12	12549	12583,99	0,28	8814,11	
7	20	12340*	12340	0	20,36	2,06	7885,9	7908,32	12338	12385,12	0,38	9266,05	
8	20	13167*	13167	0	7,08	0,31	1,09	8,48	13167*	13167	0,00	4,91	
9	20	12980*	12980	0	11,64	0,44	3,23	15,31	12980	12996,38	0,13	7561,05	
10	25	12100	12103,84	0,03	25,73	23,61	14350,66	14400	12099	12163,72	0,53	11412,01	
11	20	12912*	12912	0	13,61	0,50	5,64	19,75	12912	12931,32	0,15	8567,91	
12	20	12782*	12782	0	16,97	1,06	93,55	111,58	12782	12817,05	0,27	8754,51	
13	20	13195*	13195	0	0,48	0,03	0,19	0,70	13195*	13195	0,00	0,58	
14	20	13099*	13099,67	0,01	12,03	0,61	3,75	16,39	13099*	13099	0,00	21,28	
15	15	12686*	12686,88	0,01	20,81	2,66	2238,02	2261,49	12682	12726,83	0,35	9652,27	
16	20	13028*	13028	0	14,31	0,83	8,23	23,37	13028	13032,33	0,03	9046,44	
17	20	13175*	13175	0	1,12	0,06	0,38	1,56	13175*	13175	0,00	1,34	
18	40	12964*	12964,50	0	18,31	24,8	6062,63	6105,74	12964	12979,21	0,12	9122,36	
19	20	13129*	13129,73	0,01	2,12	0,19	15,36	17,67	13129*	13129	0,00	5,59	
20	15	13086*	13086,90	0,01	2,75	1,76	40,16	44,67	13086	13087,33	0,01	11434,7	
Média	20,75	12864,85	12865,23	0	12,25	3,11	1536,61	1551,97	12864,30	12882,59	0,15	6682,11	

*Soluções ótimas com gaps fechados.

Para as instâncias com 100 e 250 pontos, tanto o CPLEX quanto a $DC_{\alpha\beta}PMNRSC^n$ apresentaram as soluções ótimas (100% dos pontos rotulados sem conflitos) em um tempo computacional inferior a 1 segundo. Já para as instâncias com 13206 pontos, nem o CPLEX (para o PMNRSC) nem a $DC_{\alpha\beta}PMNRSC^n$ apresentaram resultados, o CPLEX devido à falta de memória, e a $DC_{\alpha\beta}PMNRSC^n$ devido à heurística de Sachdeva (2004) (utilizada na cópia dos vértices) que não foi capaz de determinar as cópias em um tempo computacional aceitável.

Nas Tabelas 6.1, 6.2, 6.3 e 6.4, a coluna n indica o número de *clusters* em que o problema foi particionado; as colunas LB e UB indicam os limitantes inferiores e superiores obtidos; a coluna *Gap* indica a diferença entre LB e UB, ou seja, $Gap = (UB - LB)/100$; a coluna M. & P. indica o tempo necessário para montar e particionar o grafo do problema, e a coluna Subp. o tempo necessário para montagem dos subproblemas; já a coluna Subg. apresenta o tempo de processamento do algoritmo de subgradientes, e a coluna Total (e Tempo - para o CPLEX) o tempo total de processamento. A última linha dessas tabelas apresentam as médias aritméticas referentes a cada coluna.

Como pode ser observado na Tabela 6.1, a $DC_{\alpha}PMCIV^n$ só não encontrou a solução ótima para a instância 10. Já o CPLEX encontrou as soluções ótimas (LB) para 15 instâncias, porém só fechou os *gaps* para 5 destas. Quanto ao tempo computacional, percebe-se claramente a eficiência da $DC_{\alpha}PMCIV^n$, que utiliza em média 1551,97 segundos para obter as soluções ótimas para praticamente todas as instâncias. Já o CPLEX, com uma média de 6682,11 segundos, não consegue provar a otimalidade para a maioria das instâncias.

Considerando a média das colunas LB (Tabela 6.1), pode-se obter o percentual de pontos rotulados, tanto pelo CPLEX quanto pela $DC_{\alpha}PMCIV^n$, considerando a relação com o número total de pontos, ou seja, $\% = (LB/N)*100$. Logo, percebe-se que o percentual médio de pontos rotulados pelo CPLEX é de 97,41%, e 97,42% pela $DC_{\alpha}PMCIV^n$. Vale lembrar que nessa abordagem (PMCIV) nem todos os pontos são rotulados, porém, aqueles que forem rotulados não apresentarão conflitos.

Tabela 6.2 - Soluções obtidas pelo PMNRSC para as instâncias com 500 pontos.

Inst.	DC _{αβ} PMNRSC ^m										CPLEX		
	n	LB	UB	Gap	M. & P.	Subp.	Subg.	Total	LB	UB	Gap	Tempo	
1	2	500*	500	0	0,03	0,01	0,32	0,36	500*	500	0	0,08	
2	2	500*	500	0	0,03	0,02	0,27	0,32	500*	500	0	0,13	
3	2	498*	498	0	0,00	0,00	0,22	0,22	498*	498	0	0,14	
4	2	500*	500	0	0,05	0,02	0,23	0,30	500*	500	0	0,06	
5	2	496*	496	0	0,02	0,00	0,56	0,58	496*	496	0	0,25	
6	2	500*	500	0	0,02	0,01	0,14	0,17	500*	500	0	0,06	
7	2	498*	498	0	0,03	0,01	0,66	0,70	498*	498	0	0,09	
8	2	498*	498	0	0,00	0,02	0,42	0,44	498*	498	0	0,11	
9	2	496*	496	0	0,03	0,00	0,95	0,98	496*	496	0	0,20	
10	2	500*	500	0	0,02	0,00	0,06	0,08	500*	500	0	0,05	
11	2	500*	500	0	0,01	0,02	0,14	0,17	500*	500	0	0,06	
12	2	500*	500	0	0,05	0,00	0,44	0,49	500*	500	0	0,06	
13	2	498*	498	0	0,01	0,00	0,39	0,40	498*	498	0	0,11	
14	2	498*	498	0	0,00	0,00	0,34	0,34	498*	498	0	0,09	
15	2	498*	498	0	0,05	0,00	0,78	0,83	498*	498	0	0,09	
16	2	496*	496	0	0,02	0,00	0,47	0,49	496*	496	0	0,42	
17	2	493*	493	0	0,05	0,00	0,94	0,99	493*	493	0	0,41	
18	2	500*	500	0	0,00	0,00	0,16	0,16	500*	500	0	0,05	
19	2	500*	500	0	0,00	0,01	0,16	0,17	500*	500	0	0,06	
20	2	500*	500	0	0,01	0,01	0,17	0,19	500*	500	0	0,08	
21	2	498*	498	0	0,06	0,00	0,77	0,83	498*	498	0	0,14	
22	2	500*	500	0	0,00	0,01	0,30	0,31	500*	500	0	0,05	
23	2	495*	495	0	0,00	0,01	0,39	0,40	495*	495	0	0,36	
24	2	498*	498	0	0,00	0,02	0,23	0,25	498*	498	0	0,08	
25	2	500*	500	0	0,05	0,00	0,09	0,14	500*	500	0	0,05	
Média	2	498,4	498,4	0	0,02	0,01	0,38	0,41	498,4	498,4	0	0,13	

*Soluções ótimas com gaps fechados.

Na Tabela 6.2, pode-se observar que tanto o CPLEX aplicado ao modelo proposto para o PMNRSC quanto a $DC_{\alpha\beta}PMNRSC^n$ encontram as soluções ótimas para todas as instâncias em um tempo inferior a 1 segundo. O percentual de pontos rotulados sem conflitos, em ambos os casos, é de 99,68%. Nessa abordagem (PMNRSC) todos os pontos são rotulados.

A Tabela 6.3 apresenta os resultados obtidos para o PMNRSC com as instâncias de 750 pontos. Nessa tabela, observa-se que mais uma vez que a $DC_{\alpha\beta}PMNRSC^n$ encontra as soluções ótimas para todas as instâncias em um tempo computacional médio de 24,31 segundos. Já o CPLEX apresenta um *gap* residual médio de 0,11%, encontrando as soluções ótimas apenas para 18 das 25 instâncias, e isso com um tempo médio de 4340,83 segundos. O percentual de pontos rotulados em ambos os casos é de 97,96%.

Já na Tabela 6.4, com os resultados obtidos para o PMNRSC com as instâncias de 1000 pontos, a $DC_{\alpha\beta}PMNRSC^n$ consegue provar a otimalidade para apenas 5 das 25 instâncias, porém apresenta um *gap* médio de apenas 0,48%. Já o CPLEX não consegue provar a otimalidade para nenhuma das 25 instâncias, e ainda apresenta um *gap* de 2,71%. Em relação ao tempo computacional, a $DC_{\alpha\beta}PMNRSC^n$ apresenta um tempo ligeiramente superior ao CPLEX. O percentual de pontos rotulados é de 93,84% e 93,74% para o CPLEX e a $DC_{\alpha\beta}PMNRSC^n$, respectivamente.

A Tabela 6.5 apresenta uma comparação entre o percentual de pontos rotulados sem conflitos obtidos por um conjunto dos principais trabalhos referentes ao PRCP. Como pode ser observado, tanto o modelo proposto (PMNRSC) quanto à decomposição $DC_{\alpha\beta}PMNRSC^n$ apresentam melhores resultados, além de provar a otimalidade para boa parte das instâncias tratadas. Percebe-se claramente uma pequena melhora entre um trabalho e outro, o que destaca ainda mais a melhora obtida pelas abordagens propostas, que são relativamente grandes em comparação com as demais.

Quanto ao tempo computacional médio, não foram colocados valores na Tabela 6.5 devido à diversidade das máquinas nas quais os experimentos foram realizados. Entretanto, deve-se deixar claro que o tempo computacional das abordagens propostas (ver Tabelas 6.2, 6.3 e 6.4) são mais elevados, o que é aceitável por se tratarem de métodos exatos para resolução do PRCP. Entretanto, como apresentado em Alvim e Taillard (2009), os tempos computacionais dos métodos heurísticos descritos nessa tabela são inferiores a 100 segundos, e aproximadamente 1 hora para os métodos propostos por Ribeiro e Lorena (2008a) e Ribeiro e Lorena (2008b).

Tabela 6.3 - Soluções obtidas pelo PMNRSC para as instâncias com 750 pontos.

Inst.	$DC_{\alpha\beta}PMNRSC^n$										CPLEX			
	n	LB	UB	Gap	M. & P.	Subp.	Subg.	Total	LB	UB	Gap	Tempo		
1	10	739*	739	0	0,08	0,02	1,00	1,10	739*	739	0	44,67		
2	10	736*	736	0	0,03	0,02	1,66	1,71	736*	736	0	2033,56		
3	10	731*	731	0	0,16	0,02	32,00	32,18	731	733,78	0,38	8957,94		
4	10	741*	741	0	0,06	0,02	1,58	1,66	741*	741	0	10,69		
5	10	739*	739	0	0,06	0,08	1,24	1,38	739*	739	0	143,73		
6	10	730*	730	0	0,06	0,01	22,75	22,82	730	731,09	0,15	15210,70		
7	10	737*	737	0	0,06	0,02	2,36	2,44	737*	737	0	70,05		
8	10	736*	736	0	0,06	0,02	1,41	1,49	736*	736	0	1913,95		
9	10	726*	726	0	0,06	0,01	8,55	8,62	726	731,21	0,71	5183,67		
10	10	743*	743	0	0,03	0,19	40,01	40,23	743*	743	0	2,13		
11	10	733*	733	0	0,16	0,02	2,47	2,65	733*	733	0	2992,88		
12	10	734*	734	0	0,06	0,01	1,80	1,87	734*	734	0	20,08		
13	10	743*	743	0	0,05	0,01	2,02	2,08	743*	743	0	1,94		
14	10	728*	728	0	0,08	0,02	5,05	5,15	728	731,09	0,42	8911,52		
15	10	730*	730	0	0,06	0,02	32,48	32,56	730*	730	0	15095,80		
16	10	729*	729	0	0,05	0,03	396,02	396,10	729*	729	0	11111,20		
17	10	729*	729	0	0,05	0,02	6,70	6,77	729	731,74	0,37	10142,30		
18	10	737*	737	0	0,08	0,02	1,59	1,69	737*	737	0	154,44		
19	10	740*	740	0	0,12	0,06	3,41	3,59	740*	740	0	9,97		
20	10	737*	737	0	0,06	0,02	2,83	2,91	737*	737	0	81,36		
21	10	731*	731	0	0,08	0,02	19,81	19,91	731	733,48	0,34	7324,94		
22	10	744*	744	0	0,05	0,02	0,84	0,91	744*	744	0	3,00		
23	10	731*	731	0	0,06	0,03	3,81	3,90	731*	731	0	36,77		
24	10	732*	732	0	0,06	0,02	8,25	8,33	732	734,95	0,40	5569,06		
25	10	732*	732	0	0,06	0,01	5,66	5,73	732*	732	0	13494,50		
Média	10	734,72	734,72	0	0,07	0,03	24,21	24,31	734,72	735,53	0,11	4340,83		

*Soluções ótimas com gaps fechados.

Tabela 6.4 - Soluções obtidas pelo PMNRSC para as instâncias com 1000 pontos.

Inst.	$DC_{\alpha\beta}PMNRSC^n$										CPLEX			
	n	LB	UB	Gap	M. & P.	Subp.	Subg.	Total	LB	UB	Gap	Tempo		
1	25	939*	939,98	0,10	0,34	0,95	1347,61	1348,90	939	961,03	2,29	8217,50		
2	25	933	939,62	0,70	0,14	1,77	14405,16	14407,07	934	962,59	2,97	9372,64		
3	20	934*	934,00	0,00	0,14	0,50	2751,06	2751,70	933	961,14	2,93	7993,61		
4	40	929	939,90	1,16	0,16	9,45	14446,88	14456,49	933	962,43	3,06	9234,03		
5	15	960	962,33	0,24	0,14	0,56	14413,56	14414,26	961	976,20	1,56	8834,52		
6	20	932*	932,92	0,10	0,16	0,16	1400,39	1400,71	932	960,52	2,97	8020,03		
7	25	928	931,99	0,43	0,14	1,02	14418,14	14419,30	928	960,06	3,34	8819,64		
8	20	940	942,21	0,23	0,14	0,77	14418,98	14419,89	939	968,44	3,04	8312,06		
9	20	923	930,32	0,79	0,14	0,77	15143,19	15144,10	925	959,52	3,60	8894,97		
10	20	943	946,48	0,37	0,14	0,70	14703,78	14704,62	944	971,54	2,83	8861,16		
11	25	947*	947,90	0,09	0,14	1,11	3332,97	3334,22	947	967,78	2,15	8521,02		
12	25	934	936,44	0,26	0,17	2,11	14562,00	14564,28	934	956,23	2,32	7860,77		
13	25	954	956,03	0,21	0,12	1,39	14528,36	14529,87	955	972,99	1,85	8964,73		
14	25	930	941,15	1,18	0,17	6,42	14523,11	14529,70	933	960,52	2,87	9293,41		
15	25	932	937,00	0,53	0,19	2,98	16450,00	16453,17	934	963,99	3,11	9248,70		
16	25	928	936,01	0,86	0,16	2,27	14450,20	14452,63	931	959,58	2,98	9647,72		
17	25	937	939,63	0,28	0,14	0,52	14467,31	14467,97	937	964,61	2,86	11243,55		
18	25	946	947,27	0,13	0,16	1,30	14507,63	14509,09	946	968,91	2,36	8934,28		
19	25	950	953,25	0,34	0,16	1,58	15262,58	15264,32	949	970,46	2,21	9214,44		
20	25	929	945,84	1,78	0,19	2,56	14416,59	14419,34	934	966,22	3,33	9253,99		
21	25	928	932,22	0,45	0,17	1,09	14435,02	14436,28	930	955,38	2,66	8707,73		
22	25	952	955,41	0,36	0,14	2,33	14403,81	14406,28	952	974,27	2,29	8638,53		
23	25	933	940,23	0,77	0,14	2,36	14608,95	14611,45	934	960,13	2,72	9103,09		
24	25	929	934,39	0,58	0,14	2,77	14461,00	14463,91	932	964,74	3,39	9178,91		
25	25	945*	945,70	0,07	0,12	0,62	3512,50	3513,24	944	964,70	2,15	8601,40		
Média	24,20	937,40	941,93	0,48	0,16	1,92	12214,83	12216,91	938,40	964,56	2,71	8918,90		

*Soluções ótimas com gaps fechados.

Tabela 6.5 - Comparação dos resultados obtidos para o PMNRSC com a literatura.

Métodos de solução	Número de pontos - N			
	250	500	750	1000
CPLEX aplicado ao modelo proposto para o PMNRSC	100	99,68*	97,96	93,84
$DC_{\alpha\beta}PMNRSC^n$	100	99,68*	97,96*	93,74
Pop(10) (ALVIM; TAILLARD, 2009)	100	99,67	97,46	91,94
Pop(30) (ALVIM; TAILLARD, 2009)	100	99,67	97,72	92,54
Pop(70) (ALVIM; TAILLARD, 2009)	100	99,67	97,73	92,58
Pop(asc) (ALVIM; TAILLARD, 2009)	100	99,67	97,72	92,68
Tabu(50n) (ALVIM; TAILLARD, 2009)	100	99,57	97,53	91,54
Tabu(100n) (ALVIM; TAILLARD, 2009)	100	99,57	97,54	91,54
Tabu(500n) (ALVIM; TAILLARD, 2009)	100	99,57	97,55	91,59
Geração de Colunas (RIBEIRO; LORENA, 2008a)	100	99,67	97,67	92,40
LagClus (RIBEIRO; LORENA, 2008b)	100	99,67	97,65	91,42
GRASP(6) (CRAVO et al., 2008)	100	99,67	97,72	92,20
GRASP(5) (CRAVO et al., 2008)	100	99,67	97,70	92,02
AGC (YAMAMOTO; LORENA, 2005)	100	99,60	97,10	90,70
Busca Tabu (YAMAMOTO et al., 2002)	100	99,26	96,76	90,00
Algoritmo Genético com máscaras (VERNER et al., 1997)	99,98	98,79	95,99	88,96
Algoritmo Genético (VERNER et al., 1997)	98,40	92,59	82,38	65,70
<i>Simulated Annealing</i> (CHRISTENSEN et al., 1995)	99,90	98,30	92,30	82,09
<i>3-opt Gradient Descent</i> (CHRISTENSEN et al., 1995)	99,76	97,34	89,44	77,83
<i>2-opt Gradient Descent</i> (CHRISTENSEN et al., 1995)	99,36	95,62	85,60	73,37
<i>Gradient Descent</i> (CHRISTENSEN et al., 1995)	95,47	86,46	72,40	58,29
<i>Greedy Algorithm</i> (CHRISTENSEN et al., 1995)	88,82	75,15	58,57	43,41

*Todas as soluções ótimas foram comprovadas.

6.6 Considerações finais

Como mencionado anteriormente, percebe-se claramente na literatura uma disputa acirrada sobre as melhores soluções para o PRCP. Entretanto, a maioria dos métodos utilizados são baseados em heurísticas e metaheurísticas, o que não permite verificar o quão próximo do ótimo estão as soluções obtidas.

Provavelmente, isso se deve a não disponibilidade de uma formulação linear inteira para representar o problema em sua forma mais explorada, a maximização do número de rótulos sem conflitos (PMNRSC). Logo, o modelo proposto para o PMNRSC surge como uma alternativa para auxiliar na verificação das soluções obtidas para o PRCP, e como pode ser observado nos experimentos computacionais apresentados, os resultados apresentados são excelentes.

Já a decomposição aplicada a uma formulação já existente baseada no Problema do Máximo Conjunto Independente de Vértices ($DC_{\alpha}PMCIV^n$) apresenta indícios

de ser uma excelente alternativa para resolução de problemas modelados como um $PMCIIV$. Esse fato pode ser comprovado pelos resultados obtidos para as instâncias com 13206, que são relativamente grandes, e a $DC_{\alpha}PMCIIV^n$ consegue encontrar as soluções ótimas para praticamente todas.

Além disso, uma nova alternativa para resolução do modelo proposto também é apresentada. Esse método é baseado na Decomposição Lagrangiana, que como apresentado no capítulo anterior e em outros trabalhos encontrados na literatura, apresenta bons resultados para alguns problemas modelados em grafos. Os resultados obtidos por essa abordagem ($DC_{\alpha\beta}PMNRSC^n$) apresentam melhores soluções para as instâncias consideradas nos principais trabalhos referentes ao PRCP, e o mais interessante, as soluções ótimas, algumas até então desconhecidas, são obtidas para quase todas as instâncias.

Enfim, os resultados obtidos apresentam as melhores soluções conhecidas para o PRCP, superando métodos recentes e de relevante importância para o PRCP. Além disso, a DecLag proposta ainda contribui como uma interessante ferramenta para resolução de problemas representados por grafos de conflitos.

Todos os experimentos apresentados neste capítulo foram realizados em um PC com processador *Pentium Dual Core* de 1.73 GHz e 1GB de memória RAM. Toda a implementação foi desenvolvida na linguagem C++ com chamadas às bibliotecas do *software* CPLEX 10.0 e da METIS.

O estudo apresentado neste capítulo contou com a colaboração do Prof. Dr. Glaydston Mattos Ribeiro, da Universidade Federal do Espírito Santo (UFES).

7 CONCLUSÕES E TRABALHOS FUTUROS

Esta tese constituiu na exploração de algumas técnicas baseadas em heurísticas e metaheurísticas, métodos de enumeração implícita, e relaxações e decomposições para resolução e/ou obtenção de limitantes para quatro problemas distintos. Além disso, foram abordados também a análise e o desenvolvimento de modelos matemáticos para representação desses problemas.

Em relação ao DARP, o modelo proposto se mostrou flexível para o tratamento de casos distintos do problema, permitindo a busca por diferentes tipos de soluções, principalmente aquelas que priorizam a qualidade do serviço prestado, e não somente os custos operacionais. Já a aplicação do *Simulated Annealing*, juntamente com as outras heurísticas específicas ao DARP, foi capaz de obter, em todos os casos e com pouco tempo de processamento, soluções válidas para o problema. Além disso, a estrutura de vizinhança utilizada mostrou ser adequada e eficiente para exploração do espaço de soluções. Os resultados obtidos foram comparados com duas outras abordagens recentes encontradas na literatura, e em todos os casos a “qualidade do serviço” foi expressivamente melhor, o que na prática, em se tratando da perspectiva humana, reflete melhores soluções.

Já considerando o PAB, todos os métodos propostos apresentaram bons resultados, com destaque para o ATP/PL, que superou a Busca Tabu proposta por Cordeau et al. (2005). Como verificado com o auxílio do CPLEX, que sozinho não foi capaz de resolver boa parte das instâncias, as soluções obtidas pelo ATP/PL são próximas ou iguais às ótimas. Vale destacar também que o ATP integrado à PL por meio da Geração de Colunas (ATP/PL) foi capaz de “aprimorar” o desempenho do ATP quando utilizado de forma isolada. Logo, percebe-se que o ATP/PL apresenta indícios de ser adequado à resolução de outros problemas para os quais a geração de colunas seja indicada.

Para o PQ, todas as abordagens propostas apresentam limitantes melhores que as relaxações lagrangianas tradicionais. A simples inserção da restrição de corte já apresenta um ganho considerável. De uma forma geral, as abordagens baseadas na LagClus e na DecLag apresentaram excelentes resultados para as instâncias com até 200 variáveis, com densidade variada. Já para as instâncias com 500 variáveis, tais métodos falharam, principalmente pela dificuldade encontrada pelo CPLEX em resolver os subproblemas dentro de um tempo aceitável. Nesse caso, percebe-se claramente a influência do número de arestas relaxadas, no caso da LagClus, e do número

de variáveis copiadas, no caso da DecLag, pois esses métodos encontram grandes dificuldades para problemas com alta densidade. Entretanto, esses resultados indicam uma superioridade em relação aos apresentados por outras decomposições encontradas na literatura para problemas quadráticos (BILLIONNET et al., 1999; CHARDAIRE: SUTTER, 1995), que são limitadas a problemas com até 100 variáveis.

Ainda considerando o PQ, a utilização das melhores abordagens propostas para a LagClus e DecLag em conjunto com a técnica de Geração de Colunas, com a utilização da Decomposição Dantzig-Wolfe, apresenta excelentes soluções para praticamente todas as instâncias. As soluções obtidas foram comparadas diretamente com outras apresentadas recentemente na literatura, e indicaram melhoras significativas.

Já a abordagem baseada na posiforme apresentada para o PQ foi apenas um início de uma busca por modelos que resultem em grafos mais esparsos, possibilitando assim um menor número de arestas *inter-clusters* após o particionamento. Entretanto, percebe-se nos resultados apresentados que essa abordagem realmente reduz a densidade dos grafos, porém aumenta consideravelmente o número de vértices e arestas. Apesar desses contratempos, os resultados apresentados são razoáveis. É interessante notar que a formulação posiforme obtida é um PMCIV, e o grafo resultante é um grafo de conflitos.

Em relação ao PRCP, considerando a abordagem baseada no PMNRSC, percebe-se na literatura uma hegemonia dos métodos heurísticos. Como mencionado anteriormente, provavelmente esse fato está relacionado à inexistência de uma formulação linear inteira para representar o problema. Assim, a partir do modelo proposto pôde-se verificar a qualidade das soluções obtidas por esses métodos heurísticos. Além disso, os resultados obtidos tanto pelo CPLEX quanto pela DecLag foram excelentes, sendo que a DecLag provou a otimalidade para a maioria das instâncias, apresentando melhores soluções para aquelas consideradas nos principais trabalhos referentes ao PRCP, e o mais interessante, várias soluções ótimas, algumas até então desconhecidas, foram obtidas para quase todas as instâncias.

Os resultados obtidos apresentaram as melhores soluções conhecidas para o PRCP, superando métodos recentes e de relevante importância para o PRCP. Já para o PRCP formulado como um PMCIV, os resultados também foram excelentes, o que indica que esse método poderá obter boas soluções também para outros problemas modelados como um PMCIV.

Analisando de uma forma geral as aplicações da LagClus e DecLag, percebe-se uma “facilidade” em tratar problemas cujo grafo é mais esparso, como no PRCP por exemplo, onde os pontos situados ao norte em um mapa “nunca” apresentarão conflitos com pontos situados no sul, por exemplo. Além disso, a heurística proposta por Sachdeva (2004) para copiar os vértices necessários para decomposição do problema aparentemente reduz significativamente o número de vértices copiados, e consequentemente o tamanho dos subproblemas.

A convergência na LagClus aparentemente fica amarrada ao Algoritmo de Subgradientes, e nesse caso, acredita-se que um método mais eficiente para resolver o dual lagrangiano talvez possa melhorar o desempenho. Já a convergência da DecLag, além da dependência do Algoritmo de Subgradientes, apresenta um agravante quando o número de cópias é alto (o que acontece “sempre” para problemas com grafos densos), pois dessa forma os subproblemas gerados também serão grandes, e consequentemente sua resolução também será árdua. Nesses casos, acredita-se que uma técnica mais eficiente para resolução dos subproblemas possa melhorar ainda mais o desempenho.

Enfim, o ATP/PL, por ser ainda uma metodologia recente e pouco explorada, aparenta ser eficiente para problemas de difícil solução, cujas formulações podem ser conhecidas ou não. Já para problemas cujas formulações são conhecidas, tanto a LagClus quanto a DecLag, com ou sem a Geração de Colunas, aparentam ser boas alternativas de resolução, principalmente para problemas representados por grafos.

7.1 Trabalhos futuros

Os diferentes modelos e métodos apresentados nesta tese apresentaram resultados satisfatórios para problemas teóricos e práticos. Entretanto, essa pesquisa pode ser complementada, principalmente no sentido de investigar novas aplicações. Com esse intuito, sugere-se:

- O estudo de casos reais para o DARP e o PAB, e a utilização dos métodos e modelos propostos.
- O tratamento do PAB em sua forma contínua, considerando as dimensões espaciais dos navios e dos berços.
- Consolidar a utilização do ATP/PL com sua aplicação a outros problemas.

- Integrar a heurística de Sachdeva (2004) na DecLag para o PQ, visando assim a redução do número de vértices copiados, e possivelmente a resolução de instâncias maiores e com maior densidade.
- A aplicação da Geração de Colunas com a LagClus e a DecLag ao PRCP, pois assim como apresentado para o PQ, os tempos computacionais poderão ser reduzidos significativamente.
- Concentrar esforços em formulações baseadas em posiformes para o PQ, buscando por grafos mais esparsos e com menor número de vértices e arestas, e conseqüentemente facilitando à aplicação da LagClus e da DecLag.
- Verificar a possibilidade de aplicação da LagClus e da DecLag a outros problemas cujos grafos geralmente são esparsos, ou apresentam regiões bem definidas (sub-grafos pouco conectados), como problemas de localização e alocação, por exemplo.
- E por fim, buscar métodos mais eficientes para resolução dos duais lagrangianos e dos subproblemas apresentados na LagClus e na DecLag.

REFERÊNCIAS BIBLIOGRÁFICAS

- ADAMS, W. P.; DEARING, P. M. On the equivalence between roof duality and lagrangean duality for unconstrained 0-1 quadratic programming problems. **Discrete Applied Mathematics**, v. 48, n. 1, p. 1–20, 1994. 46, 111, 112, 113, 114, 116, 119, 120
- ADAMS, W. P.; FORRESTER, R. J.; GLOVER, F. W. Comparisons and enhancement strategies for linearizing mixed 0-1 quadratic programs. **Discrete Optimization**, v. 1, n. 2, p. 99–120, 2004. 112, 113
- ALVIM, A. C. F.; TAILLARD, E. D. Popmusic for the point feature label placement. **European Journal of Operational Research**, v. 192, n. 2, p. 396–413, 2009. 162, 164, 174, 178, 181
- ATAMTÜRK, A.; NEMHAUSER, G. L.; SAVELSBERGH, M. W. P. Conflict graphs in solving integer programming problems. **European Journal of Operational Research**, v. 121, n. 1, p. 40–55, 2000. 48, 160
- BAUGH, J. W.; KAKIVAYA, D. K. R.; STONE, J. R. Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing. **Engineering Optimization**, v. 30, n. 2, p. 91–123, 1998. 55, 59, 82
- BAZARAA, M. S.; JARVIS, J. J.; SHERALI, H. D. **Linear programming and network flows**. New York: John Wiley & Sons, 1990. 34, 35, 36
- BEASLEY, J. E. Or-library: distributing test problems by electronic mail. **Journal of the Operational Research Society**, v. 44, n. 11, p. 1069–1072, 1990. 77, 137
- _____. **Heuristic algorithms for the unconstrained binary quadratic programming problem**. London: Imperial College, 1998. 36 p. (Relatório Técnico). Disponível em: <<http://people.brunel.ac.uk/~mastjjb/jeb/bqp.pdf>>. Acesso em: 25 out. 2008. 82, 111, 112, 130
- BERGVINSDOTTIR, K. B. **The genetic algorithm for solving the dial-a-ride problem**. 2004. 140 p. (IMM 2004-37). Dissertação (Master of Science in Engineering) — Technical University of Denmark (DTU), Lyngby, 2004. Disponível em: <[http:](http://)

//www2.imm.dtu.dk/pubdb/views/edoc_download.php/3395/ps/imm3395.ps>.

Acesso em: 25 out. 2008. 16, 65, 67, 68, 71, 74, 78, 82

BILLIONNET, A.; CALMELS, F. Linear programming for the 0-1 quadratic knapsack problem. **European Journal of Operational Research**, v. 92, n. 2, p. 310–325, 1996. 113, 116

BILLIONNET, A.; ELLOUMI, S. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. **Mathematical Programming**, v. 109, n. 1, p. 55–68, 2007. 111, 114, 153, 154, 155, 157

BILLIONNET, A.; FAYE, A.; SOUTIF, E. A new upper bound for the 0-1 quadratic knapsack problem. **European Journal of Operational Research**, v. 112, n. 3, p. 664–672, 1999. 52, 113, 124, 143, 156, 184

BILLIONNET, A.; SUTTER, A. Minimization of a quadratic pseudo-boolean function. **European Journal of Operational Research**, v. 78, n. 1, p. 106–115, 1994. 111, 112

BOROS, E.; CRAMA, Y.; HAMMER, P. L. Upper bounds for quadratic 0-1 maximization. **Operations Research Letters**, v. 9, n. 2, p. 73–79, 1990. 46, 112

_____. Chvatal cuts and odd cycle inequalities in quadratic 0-1 optimization.

Journal on Discrete Mathematics, v. 5, n. 2, p. 163–177, 1992. 46, 112

BOROS, E.; HAMMER, P. L.; SUN, X. **The DDT method for quadratic 0-1 minimization**. New Jersey: State University, 1989. 50 p. (Relatório Técnico). 112

CALVO, R. W.; LUIGI, F.; HAASTRUP, P.; MANIEZZO, V. A distributed geographic information system for the daily car pooling problem. **Computers & Operations Research**, v. 31, n. 13, p. 2263–2278, 2004. 57, 61

CHARDAIRE, P.; SUTTER, A. A decomposition method for quadratic zero-one programming. **Management Science**, v. 41, n. 4, p. 704–712, 1995. 51, 52, 111, 113, 124, 143, 156, 184

CHRISTENSEN, J.; MARKS, J.; SHIEBER, S. **Placing text labels on maps and diagrams**. London: Academic Press, 1993. 162, 163

_____. An empirical study of algorithms for point-feature label placement. **ACM Transactions on Graphics**, v. 14, n. 3, p. 203–232, 1995. 20, 160, 162, 163, 174, 181

- COHON, J. L. **Multiobjective programming & planning**. New York: Academic Press, 1978. 33
- CORDEAU, J. F. A branch-and-cut algorithm for the dial-a-ride problem. **Operations Research**, v. 54, n. 3, p. 573–586, 2006. 15, 60, 63, 65, 68, 70, 71, 78, 82
- CORDEAU, J. F.; GAUDIOSO, M.; LAPORTE, G.; MOCCIA, L. The service allocation problem at the gioia tauro maritime terminal. **European Journal of Operational Research**, v. 176, n. 2, p. 1167–1184, 2007. 88
- CORDEAU, J. F.; LAPORTE, G. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. **Transportation Research Part B: Methodological**, v. 37, n. 6, p. 579–594, 2003a. 60, 68, 73, 74, 77, 80, 82, 83
- _____. The dial-a-ride problem (DARP): variants, modeling issues and algorithms. **4OR: Quarterly Journal of the Belgian, French and Italian Operations Research Societies**, v. 1, n. 2, p. 89–101, 2003b. 56, 57, 61, 68
- _____. The dial-a-ride problem: models and algorithms. **Annals of Operations Research**, v. 153, n. 1, p. 29–46, 2007. 55, 56, 61
- CORDEAU, J. F.; LAPORTE, G.; LEGATO, P.; MOCCIA, L. Models and tabu search heuristics for the berth allocation problem. **Transportation Science**, v. 39, n. 4, p. 526–538, 2005. 85, 87, 88, 90, 91, 92, 102, 106, 108, 109, 110, 183
- CORDEAU, J. F.; LAPORTE, G.; POTVIN, J. Y.; SAVELSBERGH, M. W. P. Transportation on demand. In: BARNHART, C.; LAPORTE, G. (Ed.). **Handbooks in Operations Research and Management Science**. North-Holland: Elsevier, 2006. p. 429–466. 61
- CRAVO, G. L.; RIBEIRO, G. M.; LORENA, L. A. N. A greedy randomized adaptive search procedure for the point-feature cartographic label placement. **Computers & GeoSciences**, v. 34, n. 4, p. 373–386, 2008. 163, 181
- DANTZIG, G. B. **Linear programming and extensions**. Princeton: Princeton University Press, 1963. 34, 35
- DANTZIG, G. B.; WOLFE, P. Decomposition principle for linear programs. **Operations Research**, v. 8, n. 1, p. 101–111, 1960. 35

DESROSIERS, J.; DUMAS, Y.; SOUMIS, F. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. **American Journal of Mathematical and Management Sciences**, v. 6, n. 3, p. 301–325, 1986. 58

DRAGOVIC, B.; PARK, N. K.; RADMILOVIC, Z. Simulation modelling of ship-berth link with priority service. **Maritime Economics & Logistics**, v. 7, n. 4, p. 316–335, 2005. 88

DUMAS, Y.; DESROSIERS, J.; SOUMIS, F. The pickup and delivery problem with time windows. **European Journal of Operational Research**, v. 54, n. 1, p. 7–22, 1991. 58

FORMANN, M.; WAGNER, F. A packing problem with applications to lettering of maps. In: INTERNATIONAL ACM SYMPOSIUM ON COMPUTATIONAL GEOMETRY, 7., 1991, North Conway. **Proceedings...** North Conway: ACM, 1991. p. 281–288. 159

FORRESTER, R.; GREENBERG, H. Quadratic binary programming models in computational biology. **Algorithmic Operations Research**, v. 3, n. 2, p. 110–129, 2008. 111

FREEMAN, H. Computer name placement: principles and applications. In: MAGUIRE, D. J.; GOODCHILD, M. F.; RHIND, D. W. (Ed.). **Geographical information systems**. Essex: Longman, 1991. p. 445–456. 159

GAREY, M. R.; JOHNSON, D. S. **Computers and intractability: a guide to the theory of np-completeness**. San Francisco: Freeman, 1979. 86

GLOVER, F.; ALIDAEI, B.; REGO, C.; KOCHENBERGER, G. One-pass heuristics for large-scale unconstrained binary quadratic problems. **European Journal of Operational Research**, v. 137, n. 2, p. 272–287, 2002. 112, 152, 153, 155, 157

GLOVER, F.; KOCHENBERGER, G. A.; ALIDAEI, B. Adaptive memory tabu search for binary quadratic programs. **Management Science**, v. 44, n. 3, p. 336–345, 1998. 111, 112, 138, 148

GLOVER, F.; WOOLSEY, E. Converting a 0-1 polynomial programming problem to a 0-1 linear program. **Operations Research**, v. 22, n. 1, p. 180–182, 1974. 112, 113, 114

- GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear**. Rio de Janeiro: Campus, 2005. 165, 169
- GOLDBERG, D. E. **Genetic algorithms in search, optimization and machine learning**. Boston: Addison-Wesley, 1989. 41
- GUIGNARD, M. Lagrangean relaxation. **TOP**, v. 11, n. 2, p. 151–200, 2003. 48, 52
- GULATI, V. P.; GUPTA, S. K.; MITTAL, A. K. Unconstrained quadratic bivalent programming problem. **European Journal of Operational Research**, v. 15, n. 1, p. 121–125, 1984. 111, 112
- GUNTHER, H. O.; KIM, K. H. Container terminals and terminal operations. **OR Spectrum**, v. 28, n. 4, p. 437–445, 2006. 88
- HAMMER, P. L.; SIMEONE, B. Quadratic functions of binary variables. **Lecture Notes in Mathematics**, v. 1403, p. 1–56, 1989. 133, 134
- HANSEN, P.; MEYER, C. **Improved compact linearizations for the unconstrained quadratic 0-1 minimization problem**. Aceito pela revista *Discrete Applied Mathematics*, 2008. 112, 113
- HANSEN, P.; OGUZ, C.; MLADENOVIC, N. Variable neighborhood search for minimum cost berth allocation. **European Journal of Operational Research**, v. 191, n. 3, p. 636–649, 2008. 88
- HARARY, F. **Graph theory**. New York: Addison-Wesley, 1972. 161
- HELD, M.; KARP, R. M. The traveling salesman problem and minimum spanning trees. **Operations Research**, v. 18, n. 6, p. 1138–1162, 1970. 47, 53, 54
- _____. The traveling salesman problem and minimum spanning trees: part II. **Mathematical Programming**, v. 1, n. 1, p. 6–25, 1971. 47, 48
- HICKS, I. V.; WARREN, J. S.; WARRIER, D.; WILHELM, W. E. A branch-and-price approach for the maximum weight independent set problem. **Networks**, v. 46, n. 4, p. 198–209, 2005. 138
- HUANG, H. X.; PARDALOS, P. M.; PROKOPYEV, O. A. Lower bound improvement and forcing rule for quadratic binary programming. **Computational Optimization and Applications**, v. 33, n. 2, p. 187–208, 2006. 112

ILOG. **Ilog CPLEX 10.0 user's manual**. France, 2006. 478 p. 45, 78, 99, 106, 108, 138, 156

IMAI, A.; NAGAIWA, K.; CHAN, W. T. Efficient planning of berth allocation for container terminals in asia. **Journal of Advanced Transportation**, v. 31, n. 1, p. 75–94, 1997. 87

IMAI, A.; NISHIMURA, E.; PAPADIMITRIOU, S. The dynamic berth allocation problem for a container port. **Transportation Research Part B: Methodological**, v. 35, n. 4, p. 401–417, 2001. 87, 88

_____. Berth allocation with service priority. **Transportation Research Part B: Methodological**, v. 37, n. 5, p. 437–457, 2003. 87, 88

_____. Berthing ships at a multi-user container terminal with a limited quay capacity. **Transportation Research Part E: Logistics and Transportation Review**, v. 44, n. 1, p. 136–151, 2008. 86, 87

IMAI, A.; SUN, X.; NISHIMURA, E.; PAPADIMITRIOU, S. Berth allocation in a container port: using a continuous location space approach. **Transportation Research Part B: Methodological**, v. 39, n. 3, p. 199–221, 2005. 88

JAW, J.; ODoni, A. R.; PSARAFTIS, H. N.; WILSON, N. H. M. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. **Transportation Research Part B: Methodological**, v. 20, n. 3, p. 243–257, 1986. 55, 58

JIH, W. R.; KAO, C. Y.; HSU, F. Y. J. Using family competition genetic algorithm in pickup and delivery problem with time window constraints. In: **IEEE INTERNATIONAL SYMPOSIUM ON INTELLIGENT CONTROL**, 2002, Vancouver. **Proceedings...** Vancouver: IEEE, 2002. p. 496–501. 59

JORGENSEN, R. M.; LARSEN, J.; BERGVINSDOTTIR, K. B. Solving the dial-a-ride problem using genetic algorithms. **Journal of the Operational Research Society**, v. 58, n. 10, p. 1321–1331, 2007. 34, 60, 68, 77, 80, 82, 83

KARYPIS, G.; KUMAR, V. Multilevel k-way partitioning scheme for irregular graphs. **Journal of Parallel and Distributed Computing**, v. 48, n. 1, p. 96–129, 1998. 117, 135, 138, 143, 156, 174

- KIM, K. H.; MOON, K. C. Berth scheduling by simulated annealing. **Transportation Research Part B: Methodological**, v. 37, n. 6, p. 541–560, 2003. 87, 88
- KIRKPATRICK, S.; GELLAT, D. C.; VECCHI, M. P. Optimization by simulated annealing. **Science**, v. 220, n. 4598, p. 671–680, 1983. 39
- KOCHENBERGER, G. A.; GLOVER, F.; ALIDAEI, B.; REGO, C. An unconstrained quadratic binary programming approach to the vertex coloring problem. **Annals of Operations Research**, v. 139, n. 1, p. 229–241, 2005. 111
- KRARUP, J.; PRUZAN, P. A. Computer aided layout design. **Mathematical Programming Study**, v. 9, p. 75–94, 1978. 111
- LAGRÈZE, E. J.; LEBBAR, M. A column generation model for a scheduling problem with maintenance constraints. In: WALLACE, M.; CASEAU, Y.; LAGRÈZE, E. J.; SIMONIS, H.; PESANT, G. (Ed.). **Electronic notes in discrete mathematics**. [S.l.]: Elsevier, 2000. p. 429–466. 37
- LAUGHUNN, D. J. Quadratic binary programming with application to capital budgeting problems. **Operations Research**, v. 18, n. 3, p. 454–461, 1970. 111
- LEE, D.; WANG, H. Q.; MIAO, L. Quay crane scheduling with non-interference constraints in port container terminals. **Transportation Research Part E: Logistics and Transportation Review**, v. 44, n. 1, p. 124–135, 2008. 86, 88
- LEGATO, P.; MONACO, F.; TIGANI, N. Berth planning at gioia tauro's maritime terminal by logistic distribution models. In: ANNUAL CONFERENCE OF ITALIAN OPERATIONAL RESEARCH SOCIETY, 32., 2001, Cagliari. **Proceedings...** Cagliari: AIRO, 2001. 88
- LORENA, L. A. N.; LOPES, L. S. Computational experiments with genetic algorithms applied to set covering problems. **Pesquisa Operacional**, v. 16, n. 1, p. 41–53, 1996. 41
- MADSEN, O. B. G.; RAVN, H. F.; RYGAARD, J. M. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. **Annals of Operations Research**, v. 60, n. 1, p. 193–208, 1995. 58, 59
- MARKS, J.; SHIEBER, S. **The computational complexity of cartographic label placement**. Harvard: Harvard University, 1991. 28 p. (Relatório Técnico).

Disponível em:

<<http://www.eecs.harvard.edu/~shieber/papers/label.ps.gz>>. Acesso em: 25 out. 2008. 159

MAURI, G. R. **Resolução do problema de programação de tripulações de um sistema de transporte público via simulated annealing**. Ouro Preto: UFOP, 2003. 98 p. (Relatório Técnico 02-2003). 34, 39, 40, 82

MAURI, G. R. **Novas heurísticas para o problema de escalonamento de tripulações**. 2005. 105 p. (INPE-13602-TDI/1041). Dissertação (Mestrado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2005. 34, 43, 45

MAURI, G. R.; LORENA, L. A. N. Driver scheduling generation using a population training algorithm. In: BRAZILIAN SYMPOSIUM IN NEURAL NETWORKS, 8., 2004, São Luís. **Proceedings...** São Luís: SBRN, 2004a. 43, 45

_____. Método iterativo para resolução do problema de escalonamento de tripulações. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 36., 2004, São João del Rei. **Anais...** São João del Rei: SBPO, 2004b. p. 1593–1603. 43, 45

_____. A general multi-objective model for dial-a-ride problem solved by simulated annealing. In: INTERNATIONAL SYMPOSIUM ON MATHEMATICAL PROGRAMMING, 19., 2006, Rio de Janeiro. **Proceedings...** Rio de Janeiro: ISMP, 2006a. p. 108. 34, 40

_____. Simulated annealing aplicado a um modelo geral do problema de roteirização e programação de veículos. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 38., 2006, Goiânia. **Anais...** Goiânia: SBPO, 2006b. p. 1051–1065. 40, 82

_____. A new hybrid heuristic for driver scheduling. **International Journal of Hybrid Intelligent Systems**, v. 4, n. 1, p. 39–47, 2007. 34, 43, 45

_____. A lagrangean decomposition for unconstrained binary quadratic programming problem. In: LATIN-IBERO AMERICAN CONGRESS ON OPERATIONS RESEARCH, 14., 2008, Cartagena de Indias. **Proceedings...** Cartagena de Indias: CLAIO, 2008a. 52

- _____. Novos limitantes para o problema de programação quadrática binária irrestrita. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 40., 2008, João Pessoa. **Anais...** João Pessoa: SBPO, 2008b. 48
- MAURI, G. R.; OLIVEIRA, A. C. M.; LORENA, L. A. N. A hybrid column generation approach for the berth allocation problem. **Lecture Notes in Computer Science**, v. 4972, p. 110–122, 2008a. 45
- _____. Heurística baseada no simulated annealing aplicada ao problema de alocação de berços. **GEPROS - Gestão da Produção, Operações e Sistemas**, v. 1, n. 1, p. 113–127, 2008b. 40, 82
- MCBRIDE, R. D.; YORMARK, J. S. An implicit enumeration algorithm for quadratic integer programming. **Management Science**, v. 26, n. 3, p. 282–296, 1980. 111
- MELACHINOUDIS, E.; ILHAN, A. B.; MIN, H. A dial-a-ride problem for client transportation in a health-care organization. **Computers & Operations Research**, v. 34, n. 3, p. 742–759, 2007. 61
- MIGDALAS, A.; PARDALOS, P. M.; VARBRAND, P. **Multilevel Optimization: algorithms and applications**. 1. ed. Berlin: Springer, 1997. 34
- MONACO, M. F.; SAMMARRA, M. The berth allocation problem: a strong formulation solved by a lagrangean approach. **Transportation Science**, v. 41, n. 2, p. 265–280, 2007. 88
- NARCISO, M. G.; LORENA, L. A. N. Lagrangean/surrogate relaxation for generalized assignment problems. **European Journal of Operational Research**, v. 114, n. 1, p. 165–177, 1999. 53
- NISHIMURA, E.; IMAI, A.; PAPADIMITRIOU, S. Berth allocation planning in the public berth system by genetic algorithms. **European Journal of Operational Research**, v. 131, n. 2, p. 282–292, 2001. 88
- OLIVEIRA, A. C. M. **Treinamento populacional em heurísticas: aplicações em otimização**. 2002. 63 p. Proposta de Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2002. 41

OLIVEIRA, A. C. M.; LORENA, L. A. N. Sequencing cutting patterns and VLSI gates by population training algorithm. In: EURO/INFORMS JOINT INTERNATIONAL MEETING, 2003, Istanbul. **Proceedings...** Istanbul: INFORMS, 2003. 42

PALUBECKIS, G. Heuristic-based branch and bound algorithm for unconstrained quadratic 0-1 programming. **Computing**, v. 54, n. 4, p. 283–301, 1995. 112

_____. Multistart tabu search strategies for the unconstrained binary quadratic optimization problem. **Annals of Operations Research**, v. 131, n. 1, p. 259–282, 2004. 112

PAQUETTE, J.; CORDEAU, J. F.; LAPORTE, G. **Quality of service in dial-a-ride operations**. Aceito pela revista *Computers & Industrial Engineering*, 2008. 57, 61

PARDALOS, P. M.; CHAOVALITWONGSE, W.; IASEMIDIS, L. D.; SACKELLARES, J. C.; SHIAU, D. S.; CARNEY, P. R.; PROKOPYEV, O. A.; YATSENKO, V. A. Seizure warning algorithm based on optimization and nonlinear dynamics. **Mathematical Programming**, v. 101, n. 2, p. 365–385, 2004. 111

PARDALOS, P. M.; JHA, S. Complexity of uniqueness and local search in quadratic 0-1 programming. **Operations Research Letters**, v. 11, n. 2, p. 119–123, 1992. 111, 114

PARDALOS, P. M.; RODGERS, G. P. Computational aspect of a branch and bound algorithm for quadratic 0-1 programming. **Computing**, v. 45, n. 2, p. 131–144, 1990a. 111, 112, 138

_____. Parallel branch and bound algorithms for quadratic zero-one programs on the hypercube architecture. **Annals of Operations Research**, v. 22, n. 1, p. 271–292, 1990b. 111, 112

_____. A branch and bound algorithm for the maximum clique problem. **Computers & Operations Research**, v. 19, n. 5, p. 363–375, 1992. 111

PARKER, R. G.; RARDIN, R. L. **Discrete optimization**. New York: Academic Press, 1988. 53

PHILLIPS, A. T.; ROSEN, J. B. A quadratic assignment formulation of the molecular conformation problem. **Journal of Global Optimization**, v. 4, n. 2, p. 229–241, 1994. 111

PINEDO, M. **Scheduling: theory, algorithms and systems**. Englewood Cliffs: Prentice-Hall, 2001. 86

PSARAFTIS, H. N. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. **Transportation Science**, v. 14, n. 2, p. 130–154, 1980. 57

_____. An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. **Transportation Science**, v. 17, n. 3, p. 351–357, 1983. 58

REKIEK, B.; DELCHAMBRE, A.; SALEH, H. A. Handicapped person transportation: an application of the grouping genetic algorithm. **Engineering Applications of Artificial Intelligence**, v. 19, n. 5, p. 511–520, 2006. 61

RIBEIRO, G. M. **Relaxação lagrangeana com divisão em clusters para alguns problemas de otimização modelados em grafos de conflitos**. 2007. 191 p. Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2007. 35, 48, 50, 159, 161, 162, 165, 170, 174

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, 36., 2004, São João del Rei. **Anais...** São João del Rei: SBPO, 2004. 162, 163

_____. Lagrangean relaxation with clusters and column generation for the manufacturer's pallet loading problem. **Computers & Operations Research**, v. 34, n. 9, p. 2695–2708, 2007. 48

_____. Column generation approach for the point-feature cartographic label placement problem. **Journal of Combinatorial Optimization**, v. 15, n. 2, p. 147–164, 2008a. 162, 163, 174, 178, 181

_____. Lagrangean relaxation with clusters for point-feature cartographic label placement problems. **Computers & Operations Research**, v. 35, n. 7, p. 2129–2140, 2008b. 48, 159, 162, 163, 174, 178, 181

SACHDEVA, S. **Development of a branch and price approach involving vertex cloning to solve the maximum weighted independent set problem.** 2004. 48 p. Dissertação (Master of Science in Industrial Engineering) — Texas A&M University, College Station, 2004. Disponível em: <<http://txspace.tamu.edu/bitstream/handle/1969.1/3251/etd-tamu-2004C-INEN-sachdeva.pdf>>. Acesso em: 25 out. 2008. 165, 167, 176, 185, 186

SAVELSBERGH, M. W. P. The vehicle routing problem with time windows: minimizing route duration. **ORSA Journal on Computing**, v. 4, n. 2, p. 146–154, 1992. 73, 74

STEENKEN, D.; VOSS, S.; STAHLBOCK, R. Container terminal operation and operations research: a classification and literature review. **OR Spectrum**, v. 26, n. 1, p. 3–49, 2004. 86, 88

STEUER, R. E. **Multiple criteria optimization: theory, computation & application.** New York: John Wiley & Sons, 1986. 33

STRIJK, T.; VERWEIJ, B.; AARDAL, K. **Algorithms for maximum independent set applied to map labeling.** Utrecht: Universiteit Utrecht, 2000. 42 p. (Relatório Técnico). Disponível em: <<http://www.cs.uu.nl/research/techreps/repo/CS-2000/2000-22.pdf>>. Acesso em: 25 out. 2008. 111, 161, 162, 166

TAILLARD, E.; VOSS, S. Popmusic: partial optimization metaheuristic under special intensification conditions. In: RIBEIRO, C.; HANSEN, P. (Ed.). **Essays and surveys in metaheuristics.** Berlin: Springer, 2001. p. 613–629. 164

TOTH, P.; VIGO, D. Fast local search algorithms for the handicapped persons transportation problem. In: OSMAN, I. H.; KELLY, J. P. (Ed.). **Meta-heuristics: theory and applications.** Berlin: Springer, 1996. p. 677–690. 58, 59

_____. Heuristic algorithms for the handicapped persons transportation problem. **Transportation Science**, v. 31, n. 1, p. 60–71, 1997. 59

VERNER, O. V.; WAINWRIGHT, R. L.; SCHOENEFELD, D. A. Placing text labels on maps and diagrams using genetic algorithms with masking. **Journal on Computing**, v. 9, n. 3, p. 266–275, 1997. 162, 163, 181

- VIS, I. F. A.; KOSTER, R. D. Transshipment of containers at a container terminal: an overview. **European Journal of Operational Research**, v. 147, n. 1, p. 1–16, 2003. 88
- WAGNER, F.; WOLFF, A.; KAPOOR, V.; STRIJK, T. Three rules suffice for good label placement. **Algorithmica**, v. 30, n. 2, p. 334–349, 2001. 174
- WERRA, D.; HAMMER, P. L. Weighted stability number of graphs and weighted satisfiability: the two facets of pseudo-boolean optimization. **Annals of Operations Research**, v. 149, n. 1, p. 67–73, 2007. 133
- WILHELM, W. E. A technical review of column generation in integer programming. **Optimization and Engineering**, v. 2, n. 2, p. 159–200, 2001. 38
- WOLFF, A. **Automated label placement in theory and practice**. 1999. 147 p. Tese (Doctoral in Mathematik und Informatik) — Freie Universität Berlin, Berlin, 1999. Disponível em:
<<http://i11www.iti.uni-karlsruhe.de/~awolff/pub/w-alptp-99.pdf>>.
Acesso em: 25 out. 2008. 159
- WOLSEY, L. A. **Integer programming**. New York: John Wiley & Sons, 1998. 34, 35, 36, 38, 46, 48, 116, 169
- WONG, K. I.; BELL, M. G. H. Solution of the dial-a-ride problem with multidimensional capacity constraints. **International Transactions in Operational Research**, v. 13, n. 3, p. 195–208, 2006. 60
- XIANG, Z.; CHU, C.; CHEN, H. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. **European Journal of Operational Research**, v. 174, n. 2, p. 1117–1139, 2006. 60
- YAMAMOTO, M. **Uma aplicação da busca tabu ao problema de rotulação cartográfica de pontos**. 1998. 132 p. Dissertação (Mestrado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 1998. 159
- YAMAMOTO, M.; CÂMARA, G.; LORENA, L. A. N. Tabu search heuristic for point-feature cartographic label placement. **GeoInformatica**, v. 6, n. 1, p. 77–90, 2002. 162, 163, 164, 174, 181

YAMAMOTO, M.; LORENA, L. A. N. A constructive genetic approach to point-feature cartographic label placement. In: IBARAKI, T.; NONOBE, K.; YAGIURA, M. (Ed.). **Metaheuristics: progress as real problem solvers**. Berlin: Springer, 2005. p. 285–300. 162, 163, 174, 181

ZNAMENSKY, A.; CUNHA, C. B. Um modelo para o problema de roteirização e programação do transporte de deficientes. In: CONGRESSO DE PESQUISA E ENSINO EM TRANSPORTES, 13., 1999, São Carlos. **Anais...** São Carlos: ANPET, 1999. p. 59–62. 55, 56, 59

ZORASTER, S. Integer programming applied to the map label placement problem. **Cartographica**, v. 23, n. 3, p. 16–27, 1986. 161, 162

_____. The solution of large 0-1 integer programming problems encountered in automated cartography. **Operations Research**, v. 38, n. 5, p. 752–759, 1990. 161, 162, 170, 174

_____. Expert systems and the map label placement problem. **Cartographica**, v. 28, n. 1, p. 1–9, 1991. 161, 162

A APÊNDICE - EXEMPLO DE APLICAÇÃO DAS ABORDAGENS PROPOSTAS PARA O PQ

Dada a seguinte matriz $Q = [q_{ij}]_{6 \times 6}$:

$$Q = \begin{bmatrix} q_{11} & q_{21} & q_{31} & q_{41} & q_{51} & q_{61} \\ q_{12} & q_{22} & q_{32} & q_{42} & q_{52} & q_{62} \\ q_{13} & q_{23} & q_{33} & q_{43} & q_{53} & q_{63} \\ q_{14} & q_{24} & q_{34} & q_{44} & q_{54} & q_{64} \\ q_{15} & q_{25} & q_{35} & q_{45} & q_{55} & q_{65} \\ q_{16} & q_{26} & q_{36} & q_{46} & q_{56} & q_{66} \end{bmatrix} = \begin{bmatrix} 0 & 39 & -99 & 0 & 33 & 0 \\ 39 & -72 & 78 & 0 & 0 & 0 \\ -99 & 78 & 0 & -28 & 0 & 0 \\ 0 & 0 & -28 & 0 & 84 & 51 \\ 33 & 0 & 0 & 84 & 30 & 86 \\ 0 & 0 & 0 & 51 & 86 & 0 \end{bmatrix} \quad (A.1)$$

O PQ pode ser representado por um grafo G como apresentado na Figura A.1, e sua formulação (ver Equação 5.1) é dada como apresentado na Figura A.2.

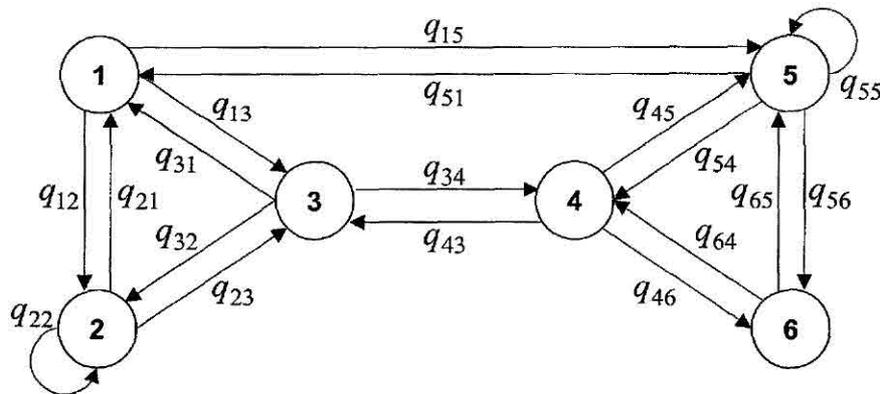


Figura A.1 - Grafo para a formulação do PQ.

PQ: $v(PQ) = \text{Maximizar:}$

$$39 x_1 x_2 - 99 x_1 x_3 + 33 x_1 x_5 - 72 x_2 + 78 x_2 x_3 - 28 x_3 x_4 + 84 x_4 x_5 + 51 x_4 x_6 + 30 x_5 + 86 x_5 x_6 + 39 x_2 x_1 - 99 x_3 x_1 + 33 x_5 x_1 + 78 x_3 x_2 - 28 x_4 x_3 + 84 x_5 x_4 + 51 x_6 x_4 + 86 x_6 x_5$$

Sujeito a:

$$x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$$

Figura A.2 - Formulação do PQ.

Considerando a simetria da matriz Q , a formulação do PQ pode ser reescrita como apresentado na Figura A.4, e seu grafo é dado pela Figura A.3;

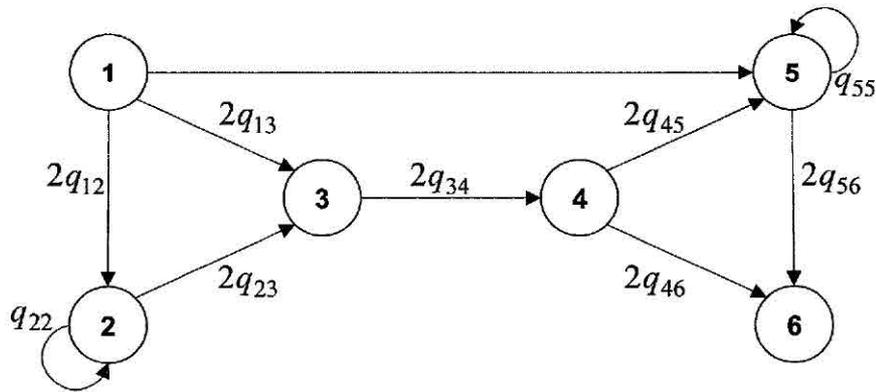


Figura A.3 - Grafo para a formulação do PQ com uma matriz simétrica.

<p>PQ: $v(PQ) = \text{Maximizar:}$</p> $-72 x_2 + 30 x_5 + 78 x_1 x_2 - 198 x_1 x_3 + 66 x_1 x_5 + 156 x_2 x_3 - 56 x_3 x_4 + 168 x_4 x_5 + 102 x_4 x_6 + 172 x_5 x_6$ <p>Sujeito a:</p> $x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$
--

Figura A.4 - Formulação do PQ com uma matriz simétrica.

Aplicando a linearização no modelo apresentado na Figura A.4, tem-se o modelo linear (PQL) equivalente apresentado na Figura A.5.

PQL: $v(PQL) = \text{Maximizar:}$

$$-72 x_2 + 30 x_5 + 78 w_{12} - 198 w_{13} + 66 w_{15} + 156 w_{23} - 56 w_{34} + 168 w_{45} + 102 w_{46} + 172 w_{56}$$

Sujeito a:

$$w_{12} - x_1 \leq 0$$

$$w_{15} - x_1 \leq 0$$

$$w_{23} - x_2 \leq 0$$

$$w_{45} - x_4 \leq 0$$

$$w_{46} - x_4 \leq 0$$

$$w_{56} - x_5 \leq 0$$

$$w_{12} - x_2 \leq 0$$

$$w_{15} - x_5 \leq 0$$

$$w_{23} - x_3 \leq 0$$

$$w_{45} - x_5 \leq 0$$

$$w_{46} - x_6 \leq 0$$

$$w_{56} - x_6 \leq 0$$

$$x_1 + x_3 - w_{13} \leq 1$$

$$x_3 + x_4 - w_{34} \leq 1$$

$$w_{13} \geq 0$$

$$w_{34} \geq 0$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$$

Figura A.5 - Formulação linear do PQ (PQL).

A.1 Limitantes para o PQ

Para o modelo PQL apresentado na Figura A.5, pode-se obter o limitante *roof dual* por meio da simples relaxação linear das variáveis de decisão, gerando assim o modelo apresentado na Figura A.6.

$$\begin{aligned} \text{roof dual} = \text{Maximizar:} \\ -72 x_2 + 30 x_5 + 78 w_{12} - 198 w_{13} + 66 w_{15} + 156 w_{23} - 56 w_{34} + 168 w_{45} \\ + 102 w_{46} + 172 w_{56} \end{aligned}$$

Sujeito a:

$$\begin{aligned} w_{12} - x_1 &\leq 0 \\ w_{15} - x_1 &\leq 0 \\ w_{23} - x_2 &\leq 0 \\ w_{45} - x_4 &\leq 0 \\ w_{46} - x_4 &\leq 0 \\ w_{56} - x_5 &\leq 0 \\ w_{12} - x_2 &\leq 0 \\ w_{15} - x_5 &\leq 0 \\ w_{23} - x_3 &\leq 0 \\ w_{45} - x_5 &\leq 0 \\ w_{46} - x_6 &\leq 0 \\ w_{56} - x_6 &\leq 0 \\ x_1 + x_3 - w_{13} &\leq 1 \\ x_3 + x_4 - w_{34} &\leq 1 \\ w_{13} &\geq 0 \\ w_{34} &\geq 0 \\ 0 \leq x_1 &\leq 1 \\ 0 \leq x_2 &\leq 1 \\ 0 \leq x_3 &\leq 1 \\ 0 \leq x_4 &\leq 1 \\ 0 \leq x_5 &\leq 1 \\ 0 \leq x_6 &\leq 1 \end{aligned}$$

Figura A.6 - Relaxação linear do PQL (*roof dual*).

Já o modelo baseado na restrição de corte descrito na Seção 5.4 é apresentado na Figura A.7.

<p>PQLC: $v(PQLC) = \text{Maximizar:}$</p> $-72 x_2 + 30 x_5 + 78 w_{12} - 198 w_{13} + 66 w_{15} + 156 w_{23} - 56 w_{34} + 168 w_{45} + 102 w_{46} + 172 w_{56}$ <p>Sujeito a:</p> $w_{12} - x_1 \leq 0$ $w_{15} - x_1 \leq 0$ $w_{23} - x_2 \leq 0$ $w_{45} - x_4 \leq 0$ $w_{46} - x_4 \leq 0$ $w_{56} - x_5 \leq 0$ $w_{12} - x_2 \leq 0$ $w_{15} - x_5 \leq 0$ $w_{23} - x_3 \leq 0$ $w_{45} - x_5 \leq 0$ $w_{46} - x_6 \leq 0$ $w_{56} - x_6 \leq 0$ $x_1 + x_3 - w_{13} \leq 1$ $x_3 + x_4 - w_{34} \leq 1$ $w_{13} \geq 0$ $w_{34} \geq 0$ $x_1 + x_2 + x_3 - w_{12} - w_{23} - w_{13} \leq 1$ $x_4 + x_5 + x_6 - w_{45} - w_{56} - w_{46} \leq 1$ $0 \leq x_1 \leq 1$ $0 \leq x_2 \leq 1$ $0 \leq x_3 \leq 1$ $0 \leq x_4 \leq 1$ $0 \leq x_5 \leq 1$ $0 \leq x_6 \leq 1$

Figura A.7 - Formulação do PQLC.

Os modelos resultantes da Relaxação Lagrangiana tradicional aplicada às restrições 5.4 e 5.6, descritas na Seção 5.3, são apresentados nas Figuras A.8 e A.9, respectivamente. A idéia das demais relaxações descritas na Seção 5.3 é semelhante.

$L_{\alpha}PQL$: $v(L_{\alpha}PQL) = \text{Maximizar:}$

$$-72 x_2 + 30 x_5 + 78 w_{12} - 198 w_{13} + 66 w_{15} + 156 w_{23} - 56 w_{34} + 168 w_{45} \\ + 102 w_{46} + 172 w_{56} - \alpha_{12}(w_{12} - x_1) - \alpha_{15}(w_{15} - x_1) - \alpha_{23}(w_{23} - x_2) \\ - \alpha_{45}(w_{45} - x_4) - \alpha_{46}(w_{46} - x_4) - \alpha_{56}(w_{56} - x_5)$$

Sujeito a:

$$w_{12} - x_2 \leq 0$$

$$w_{15} - x_5 \leq 0$$

$$w_{23} - x_3 \leq 0$$

$$w_{45} - x_5 \leq 0$$

$$w_{46} - x_6 \leq 0$$

$$w_{56} - x_6 \leq 0$$

$$x_1 + x_3 - w_{13} \leq 1$$

$$x_3 + x_4 - w_{34} \leq 1$$

$$w_{13} \geq 0$$

$$w_{34} \geq 0$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$$

Figura A.8 - Modelo da Relaxação Lagrangiana da restrição 5.4 do PQL.

$L_{\lambda}PQL$: $v(L_{\lambda}PQL) = \text{Maximizar:}$

$$-72 x_2 + 30 x_5 + 78 w_{12} - 198 w_{13} + 66 w_{15} + 156 w_{23} - 56 w_{34} + 168 w_{45} \\ + 102 w_{46} + 172 w_{56} - \lambda_{13}(x_1 + x_3 - w_{13} - 1) - \lambda_{34}(x_3 + x_4 - w_{34} - 1)$$

Sujeito a:

$$w_{12} - x_1 \leq 0$$

$$w_{15} - x_1 \leq 0$$

$$w_{23} - x_2 \leq 0$$

$$w_{45} - x_4 \leq 0$$

$$w_{46} - x_4 \leq 0$$

$$w_{56} - x_5 \leq 0$$

$$w_{12} - x_2 \leq 0$$

$$w_{15} - x_5 \leq 0$$

$$w_{23} - x_3 \leq 0$$

$$w_{45} - x_5 \leq 0$$

$$w_{46} - x_6 \leq 0$$

$$w_{56} - x_6 \leq 0$$

$$w_{13}, w_{34} \in \{0,1\}$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$$

Figura A.9 - Modelo da Relaxação Lagrangiana da restrição 5.6 do PQL.

A.2 Relaxação Lagrangiana com *Clusters*

Considerando o particionamento do problema em 2 *clusters* ($n = 2$), como apresentado na Figura A.10, o modelo PQL^n é dado na Figura A.11.

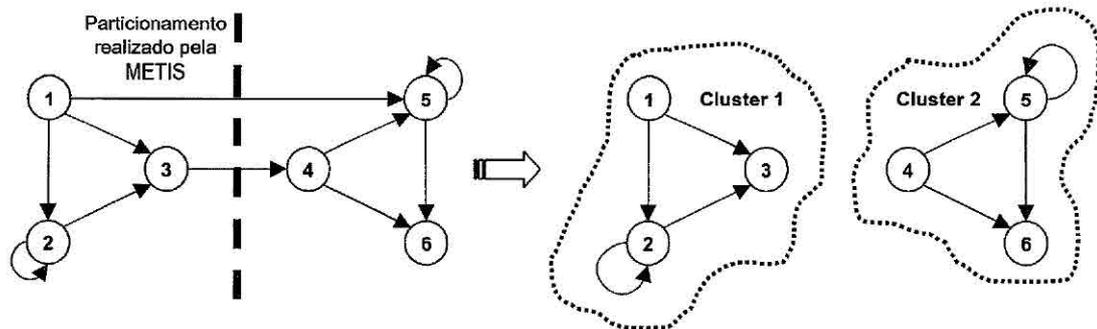


Figura A.10 - Particionamento do grafo nas abordagens da LagClus.

$$\begin{aligned}
 &PQL^2: \quad v(PQL^2) = \text{Maximizar:} \\
 &\quad -72 x_2 + 78 w_{12} - 198 w_{13} + 66 w_{15} + 156 w_{23} - 56 w_{34} \quad \{ \text{Cluster 1} \} \\
 &\quad + 30 x_5 + 168 w_{45} + 102 w_{46} + 172 w_{56} \quad \{ \text{Cluster 2} \} \\
 &\text{Sujeito a:} \\
 &\quad \left. \begin{aligned} &w_{12} - x_1 \leq 0 \\ &w_{23} - x_2 \leq 0 \\ &w_{12} - x_2 \leq 0 \\ &w_{23} - x_3 \leq 0 \\ &x_1 + x_3 - w_{13} \leq 1 \\ &w_{13} \geq 0 \end{aligned} \right\} \text{Cluster 1} \\
 &\quad \left. \begin{aligned} &w_{45} - x_4 \leq 0 \\ &w_{46} - x_4 \leq 0 \\ &w_{56} - x_5 \leq 0 \\ &w_{45} - x_5 \leq 0 \\ &w_{46} - x_6 \leq 0 \\ &w_{56} - x_6 \leq 0 \end{aligned} \right\} \text{Cluster 2} \\
 &\quad \left. \begin{aligned} &w_{15} - x_1 \leq 0 \\ &w_{15} - x_5 \leq 0 \\ &x_3 + x_4 - w_{34} \leq 1 \\ &w_{34} \geq 0 \end{aligned} \right\} \text{Inter-clusters} \\
 &\quad x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}
 \end{aligned}$$

Figura A.11 - Modelo PQL^2 utilizado nas abordagens da LagClus.

A.2.1 1ª abordagem

Relaxando as restrições *inter-clusters* (ver Figura A.11) no sentido lagrangiano, tem-se o modelo $LC_{\alpha\beta\lambda}PQL^2$, apresentado na Figura A.12. Esse modelo pode ser dividido em dois ($n = 2$) subproblemas independentes, como apresentado na Figura A.13.

<p>$LC_{\alpha\beta\lambda}PQL^2: v(LC_{\alpha\beta\lambda}PQL^2) = \text{Maximizar:}$</p> $-72 x_2 + 78 w_{12} - 198 w_{13} + 66 w_{15} + 156 w_{23} - 56 w_{34}$ $+ 30 x_5 + 168 w_{45} + 102 w_{46} + 172 w_{56}$ $- \alpha_{15}(w_{15} - x_1) - \beta_{15}(w_{15} - x_5) - \lambda_{34}(x_3 + x_4 - w_{34} - 1)$ <p>Sujeito a:</p> $w_{12} - x_1 \leq 0$ $w_{23} - x_2 \leq 0$ $w_{12} - x_2 \leq 0$ $w_{23} - x_3 \leq 0$ $x_1 + x_3 - w_{13} \leq 1$ $w_{13} \geq 0$ $w_{45} - x_4 \leq 0$ $w_{46} - x_4 \leq 0$ $w_{56} - x_5 \leq 0$ $w_{45} - x_5 \leq 0$ $w_{46} - x_6 \leq 0$ $w_{56} - x_6 \leq 0$ $w_{15}, w_{34} \in \{0,1\}$ $x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$
--

Figura A.12 - Modelo $LC_{\alpha\beta\lambda}PQL^2$ gerado na 1ª abordagem da LagClus.

<p>$LC_{\alpha\beta\lambda}PQL_1: v(LC_{\alpha\beta\lambda}PQL_1) = \text{Maximizar:}$</p> $-72 x_2 + 78 w_{12} - 198 w_{13}$ $+ (66 - \alpha_{15} - \beta_{15}) w_{15} + 156 w_{23}$ $+ (-56 + \lambda_{34}) w_{34} + \alpha_{15} x_1 - \lambda_{34} x_3$ <p>Sujeito a:</p> $w_{12} - x_1 \leq 0$ $w_{23} - x_2 \leq 0$ $w_{12} - x_2 \leq 0$ $w_{23} - x_3 \leq 0$ $x_1 + x_3 - w_{13} \leq 1$ $w_{13} \geq 0$ $w_{15}, w_{34} \in \{0,1\}$ $x_1, x_2, x_3 \in \{0,1\}$	<p>$LC_{\alpha\beta\lambda}PQL_2: v(LC_{\alpha\beta\lambda}PQL_2) = \text{Maximizar:}$</p> $+ 30 x_5 + 168 w_{45} + 102 w_{46}$ $+ 172 w_{56} + \beta_{15} x_5 - \lambda_{34} x_4$ <p>Sujeito a:</p> $w_{45} - x_4 \leq 0$ $w_{46} - x_4 \leq 0$ $w_{56} - x_5 \leq 0$ $w_{45} - x_5 \leq 0$ $w_{46} - x_6 \leq 0$ $w_{56} - x_6 \leq 0$ $x_4, x_5, x_6 \in \{0,1\}$
---	--

Figura A.13 - Subproblemas gerados na 1ª abordagem da LagClus.

A.2.2 2ª abordagem

Como descrito na Subseção 5.5.2, a 2ª abordagem não relaxa a restrição 5.15. Logo, tem-se o modelo $LC_{\beta\lambda}PQL^2$, apresentado na Figura A.14. Os subproblemas gerados são apresentados na Figura A.15.

<p>$LC_{\beta\lambda}PQL^2: v(LC_{\beta\lambda}PQL^2) = \text{Maximizar:}$</p> $-72 x_2 + 78 w_{12} - 198 w_{13} + 66 w_{15} + 156 w_{23} - 56 w_{34}$ $+ 30 x_5 + 168 w_{45} + 102 w_{46} + 172 w_{56}$ $- \beta_{15}(w_{15} - x_5) - \lambda_{34}(x_3 + x_4 - w_{34} - 1)$ <p>Sujeito a:</p> $w_{12} - x_1 \leq 0$ $w_{23} - x_2 \leq 0$ $w_{12} - x_2 \leq 0$ $w_{23} - x_3 \leq 0$ $x_1 + x_3 - w_{13} \leq 1$ $w_{13} \geq 0$ $w_{45} - x_4 \leq 0$ $w_{46} - x_4 \leq 0$ $w_{56} - x_5 \leq 0$ $w_{45} - x_5 \leq 0$ $w_{46} - x_6 \leq 0$ $w_{56} - x_6 \leq 0$ $w_{15} - x_1 \leq 0$ $w_{34} \in \{0,1\}$ $x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$
--

Figura A.14 - Modelo $LC_{\beta\lambda}PQL^2$ gerado na 2ª abordagem da LagClus.

<p>$LC_{\beta\lambda}PQL_1: v(LC_{\beta\lambda}PQL_1) = \text{Maximizar:}$</p> $-72 x_2 + 78 w_{12} - 198 w_{13}$ $+ (66 - \beta_{15}) w_{15} + 156 w_{23}$ $+ (-56 + \lambda_{34}) w_{34} - \alpha_{15} x_1 - \lambda_{34} x_3$ <p>Sujeito a:</p> $w_{12} - x_1 \leq 0$ $w_{23} - x_2 \leq 0$ $w_{12} - x_2 \leq 0$ $w_{23} - x_3 \leq 0$ $x_1 + x_3 - w_{13} \leq 1$ $w_{13} \geq 0$ $w_{15} - x_1 \leq 0$ $w_{34} \in \{0,1\}$ $x_1, x_2, x_3 \in \{0,1\}$	<p>$LC_{\beta\lambda}PQL_2: v(LC_{\beta\lambda}PQL_2) = \text{Maximizar:}$</p> $+ 30 x_5 + 168 w_{45} + 102 w_{46}$ $+ 172 w_{56} + \beta_{15} x_5 - \lambda_{34} x_4$ <p>Sujeito a:</p> $w_{45} - x_4 \leq 0$ $w_{46} - x_4 \leq 0$ $w_{56} - x_5 \leq 0$ $w_{45} - x_5 \leq 0$ $w_{46} - x_6 \leq 0$ $w_{56} - x_6 \leq 0$ $x_4, x_5, x_6 \in \{0,1\}$
---	--

Figura A.15 - Subproblemas gerados na 2ª abordagem da LagClus.

A.3 Decomposição Lagrangiana

A.3.1 1ª abordagem

Considerando o mesmo particionamento apresentado na Figura A.10 e as cópias das variáveis, de acordo com o descrito na Subseção 5.6.1, tem-se o grafo apresentado na Figura A.16, e o modelo PQL^n é dado na Figura A.17.

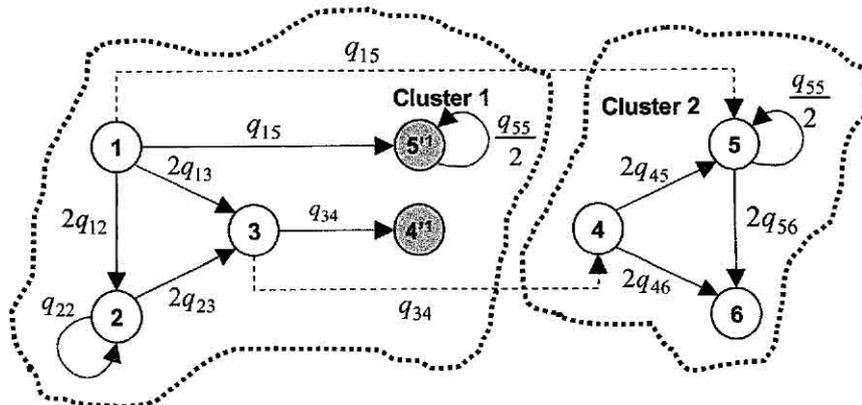


Figura A.16 - Particionamento do grafo na 1ª abordagem da DecLag.

No grafo acima, os vértices em cinza são as cópias necessárias para implementação da DecLag. Note que os coeficientes das arestas que ligam esses vértices são divididos à metade, pois a outra metade fica nas “arestas” pontilhadas. Na verdade, essas arestas pontilhadas deixam de existir, porém seus valores são considerados nas variáveis w_{15} e w_{34} apresentadas na Figura A.17.

Relaxando as restrições de cópia (ver Figura A.17) no sentido lagrangiano, tem-se o modelo $DC_{\alpha\beta}PQL^2$, apresentado na Figura A.18. Logo, os subproblemas gerados são apresentados na Figura A.19.

*PQL*²: $v(PQL^2) = \text{Maximizar:}$

$$\begin{aligned}
 & -72 x_2 + 0 x_4^1 + 15 x_5^1 + 78 w_{12} \\
 & -198 w_{13} + 33 w_{15} + 156 w_{23} \\
 & -28 w_{34} + 33 w_{15} - 28 w_{34} \\
 & + 15 x_5 + 168 w_{45} + 102 w_{46} + 172 w_{56}
 \end{aligned}
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{Cluster 1} \\ \text{Cluster 2} \end{array}$$

Sujeito a:

$$\begin{aligned}
 & w_{12} - x_1 \leq 0 \\
 & w_{23} - x_2 \leq 0 \\
 & w_{12} - x_2 \leq 0 \\
 & w_{23} - x_3 \leq 0 \\
 & x_1 + x_3 - w_{13} \leq 1 \\
 & w_{13} \geq 0
 \end{aligned}
 \left. \right\} \text{Cluster 1}$$

$$\begin{aligned}
 & w_{45} - x_4 \leq 0 \\
 & w_{46} - x_4 \leq 0 \\
 & w_{56} - x_5 \leq 0 \\
 & w_{45} - x_5 \leq 0 \\
 & w_{46} - x_6 \leq 0 \\
 & w_{56} - x_6 \leq 0
 \end{aligned}
 \left. \right\} \text{Cluster 2}$$

$$\begin{aligned}
 & w_{15} - x_1 \leq 0 \\
 & w_{15} - x_5^1 \leq 0 \\
 & x_3 + x_4^1 - w_{34} \leq 1 \\
 & w_{34} \geq 0
 \end{aligned}
 \left. \right\} \text{Inter-clusters}$$

$$\begin{aligned}
 & x_4 = x_4^1 \\
 & x_5 = x_5^1 \\
 & w_{15} = w_{15} \\
 & w_{34} = w_{34}
 \end{aligned}
 \left. \right\} \text{Restrições de cópia}$$

$x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$

Figura A.17 - Modelo *PQL*² utilizado na 1ª abordagem da DecLag.

$DC_{\alpha\beta}PQL^2$: $v(DC_{\alpha\beta}PQL^2) = \text{Maximizar:}$

$$\begin{aligned}
& -72 x_2 + 0 x^{1_4'} + 15 x^{1_5'} + 78 w_{12} - 198 w_{13} + 33 w_{15} + 156 w_{23} - 28 w_{34} \\
& + 33 w_{15'} - 28 w_{34'} + 15 x_5 + 168 w_{45} + 102 w_{46} + 172 w_{56} - \alpha^1_4(x_4 - x^{1_4'}) \\
& - \alpha^1_5(x_5 - x^{1_5'}) - \beta_{15}(w_{15} - w_{15'}) - \beta_{34}(w_{34} - w_{34'})
\end{aligned}$$

Sujeito a:

$$\begin{aligned}
& w_{12} - x_1 \leq 0 \\
& w_{23} - x_2 \leq 0 \\
& w_{12} - x_2 \leq 0 \\
& w_{23} - x_3 \leq 0 \\
& x_1 + x_3 - w_{13} \leq 1 \\
& w_{13} \geq 0 \\
& w_{45} - x_4 \leq 0 \\
& w_{46} - x_4 \leq 0 \\
& w_{56} - x_5 \leq 0 \\
& w_{45} - x_5 \leq 0 \\
& w_{46} - x_6 \leq 0 \\
& w_{56} - x_6 \leq 0 \\
& w_{15'} - x_1 \leq 0 \\
& w_{15'} - x^{1_5'} \leq 0 \\
& x_3 + x^{1_4'} - w_{34'} \leq 1 \\
& w_{34}, w_{34'} \in \{0,1\} \\
& x^{1_4'}, x^{1_5'} \in \{0,1\} \\
& x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}
\end{aligned}$$

Figura A.18 - Modelo $DC_{\alpha\beta}PQL^2$ gerado na 1ª abordagem da DecLag.

<p>$DC_{\alpha\beta}PQL_1$: $v(DC_{\alpha\beta}PQL_1) = \text{Maximizar:}$</p> $ \begin{aligned} & -72 x_2 + \alpha^1_4 x^{1_4'} + (15 + \alpha^1_5) x^{1_5'} \\ & + 78 w_{12} - 198 w_{13} + (33 - \beta_{15}) w_{15} \\ & + 156 w_{23} + (-28 - \beta_{34}) w_{34} + (33 + \beta_{15}) w_{15'} + (-28 + \beta_{34}) w_{34'} \end{aligned} $ <p>Sujeito a:</p> $ \begin{aligned} & w_{12} - x_1 \leq 0 \\ & w_{23} - x_2 \leq 0 \\ & w_{12} - x_2 \leq 0 \\ & w_{23} - x_3 \leq 0 \\ & x_1 + x_3 - w_{13} \leq 1 \\ & w_{13} \geq 0 \\ & w_{15'} - x_1 \leq 0 \\ & w_{15'} - x^{1_5'} \leq 0 \\ & x_3 + x^{1_4'} - w_{34'} \leq 1 \\ & w_{34}, w_{34'} \in \{0,1\} \\ & x^{1_4'}, x^{1_5'} \in \{0,1\} \\ & x_1, x_2, x_3 \in \{0,1\} \end{aligned} $	<p>$DC_{\alpha\beta}PQL_2$: $v(DC_{\alpha\beta}PQL_2) = \text{Maximizar:}$</p> $ \begin{aligned} & -\alpha^1_4 x_4 + (15 - \alpha^1_5) x_5 + 168 w_{45} \\ & + 102 w_{46} + 172 w_{56} \end{aligned} $ <p>Sujeito a:</p> $ \begin{aligned} & w_{45} - x_4 \leq 0 \\ & w_{46} - x_4 \leq 0 \\ & w_{56} - x_5 \leq 0 \\ & w_{45} - x_5 \leq 0 \\ & w_{46} - x_6 \leq 0 \\ & w_{56} - x_6 \leq 0 \\ & x_4, x_5, x_6 \in \{0,1\} \end{aligned} $
--	--

Figura A.19 - Subproblemas gerados na 1ª abordagem da DecLag.

A.3.2 2ª abordagem

Mais uma vez, considerando o mesmo particionamento apresentado na Figura A.10 e as cópias das variáveis, de acordo com o descrito na Subseção 5.6.2, tem-se o grafo apresentado na Figura A.20, e o modelo PQL^n é dado na Figura A.21.

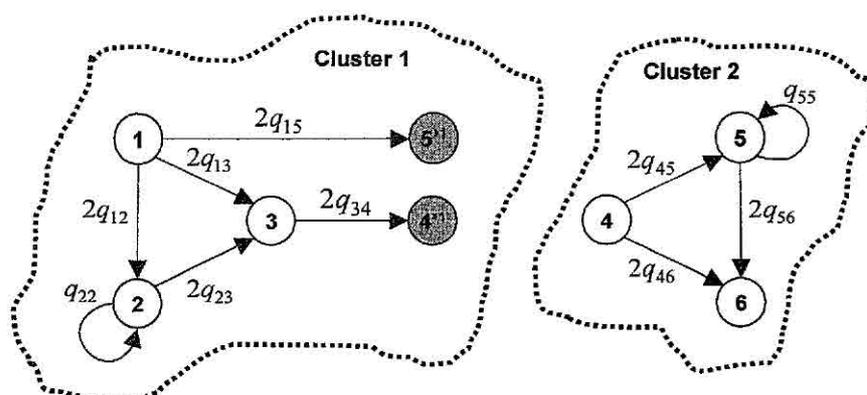


Figura A.20 - Particionamento do grafo na 2ª abordagem da DecLag.

No grafo acima, os vértices em cinza são as cópias necessárias para implementação da DecLag. Note que nesse caso os coeficientes das arestas que ligam esses vértices não são divididos.

Relaxando as restrições de cópia (ver Figura A.21) no sentido lagrangiano, tem-se o modelo $DC_\alpha PQL^2$, apresentado na Figura A.22. Logo, os subproblemas gerados são apresentados na Figura A.23.

PQL^2 : $v(PQL^2) = \text{Maximizar:}$

$$\begin{aligned}
 & -72 x_2 + 78 w_{12} - 198 w_{13} + 156 w_{23} + 66 w_{15'} - 56 w_{34'} & \{ \text{Cluster 1} \} \\
 & + 30 x_5 + 168 w_{45} + 102 w_{46} + 172 w_{56} & \{ \text{Cluster 2} \}
 \end{aligned}$$

Sujeito a:

$w_{12} - x_1 \leq 0$	}	<i>Cluster 1</i>
$w_{23} - x_2 \leq 0$		
$w_{12} - x_2 \leq 0$		
$w_{23} - x_3 \leq 0$		
$x_1 + x_3 - w_{13} \leq 1$		
$w_{13} \geq 0$		
$w_{45} - x_4 \leq 0$	}	<i>Cluster 2</i>
$w_{46} - x_4 \leq 0$		
$w_{56} - x_5 \leq 0$		
$w_{45} - x_5 \leq 0$		
$w_{46} - x_6 \leq 0$		
$w_{56} - x_6 \leq 0$		
$w_{15'} - x_1 \leq 0$	}	<i>Inter-clusters</i>
$w_{15'} - x_{5'}^1 \leq 0$		
$x_3 + x_{4'}^1 - w_{34'} \leq 1$		
$w_{34'} \geq 0$		
$x_4 = x_{4'}^1$	}	<i>Restrições de cópia</i>
$x_5 = x_{5'}^1$		

$x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$

Figura A.21 - Modelo PQL^2 utilizado na 2ª abordagem da DecLag.

$DC_{\alpha}PQL^2$: $v(DC_{\alpha}PQL^2) = \text{Maximizar:}$

$$-72 x_2 + 78 w_{12} - 198 w_{13} + 156 w_{23} + 66 w_{15'} - 56 w_{34'} \\ + 30 x_5 + 168 w_{45} + 102 w_{46} + 172 w_{56} - \alpha^1_4(x_4 - x^1_{4'}) - \alpha^1_5(x_5 - x^1_{5'})$$

Sujeito a:

$$w_{12} - x_1 \leq 0 \\ w_{23} - x_2 \leq 0 \\ w_{12} - x_2 \leq 0 \\ w_{23} - x_3 \leq 0 \\ x_1 + x_3 - w_{13} \leq 1 \\ w_{13} \geq 0 \\ w_{45} - x_4 \leq 0 \\ w_{46} - x_4 \leq 0 \\ w_{56} - x_5 \leq 0 \\ w_{45} - x_5 \leq 0 \\ w_{46} - x_6 \leq 0 \\ w_{56} - x_6 \leq 0 \\ w_{15'} - x_1 \leq 0 \\ w_{15'} - x^1_{5'} \leq 0 \\ x_3 + x^1_{4'} - w_{34'} \leq 1 \\ w_{34'} \geq 0 \\ x^1_{4'}, x^1_{5'} \in \{0,1\} \\ x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$$

Figura A.22 - Modelo $DC_{\alpha}PQL^2$ gerado na 2ª abordagem da DecLag.

$DC_{\alpha}PQL_1$: $v(DC_{\alpha}PQL_1) = \text{Maximizar:}$

$$-72 x_2 + \alpha^1_4 x^1_{4'} + (30 + \alpha^1_5) x^1_{5'} \\ + 78 w_{12} - 198 w_{13} + 156 w_{23} \\ + 66 w_{15'} - 56 w_{34'}$$

Sujeito a:

$$w_{12} - x_1 \leq 0 \\ w_{23} - x_2 \leq 0 \\ w_{12} - x_2 \leq 0 \\ w_{23} - x_3 \leq 0 \\ x_1 + x_3 - w_{13} \leq 1 \\ w_{13} \geq 0 \\ w_{15'} - x_1 \leq 0 \\ w_{15'} - x^1_{5'} \leq 0 \\ x_3 + x^1_{4'} - w_{34'} \leq 1 \\ w_{34'} \geq 0 \\ x^1_{4'}, x^1_{5'} \in \{0,1\} \\ x_1, x_2, x_3 \in \{0,1\}$$

$DC_{\alpha}PQL_2$: $v(DC_{\alpha}PQL_2) = \text{Maximizar:}$

$$-\alpha^1_4 x_4 + (30 - \alpha^1_5) x_5 + 168 w_{45} \\ + 102 w_{46} + 172 w_{56}$$

Sujeito a:

$$w_{45} - x_4 \leq 0 \\ w_{46} - x_4 \leq 0 \\ w_{56} - x_5 \leq 0 \\ w_{45} - x_5 \leq 0 \\ w_{46} - x_6 \leq 0 \\ w_{56} - x_6 \leq 0 \\ x_4, x_5, x_6 \in \{0,1\}$$

Figura A.23 - Subproblemas gerados na 2ª abordagem da DecLag.

A.3.3 3ª abordagem

Essa abordagem utiliza todos os elementos da matriz Q . Logo, o modelo linear utilizado (PQLT) é apresentado na Figura A.24, e o respectivo grafo é o mesmo apresentado na Figura A.1.

$$\begin{aligned} \text{PQLT: } v(\text{PQLT}) = \text{Maximizar:} \\ & -72 x_2 + 30 x_5 + 39 w_{12} - 99 w_{13} + 33 w_{15} + 39 w_{21} + 78 w_{23} - 99 w_{31} \\ & + 78 w_{32} - 28 w_{34} - 28 w_{43} + 84 w_{45} + 51 w_{46} + 33 w_{51} + 84 w_{54} + 86 w_{56} \\ & + 51 w_{64} + 86 w_{65} \\ \text{Sujeito a:} \\ & w_{12} - x_1 \leq 0 \\ & w_{15} - x_1 \leq 0 \\ & w_{21} - x_2 \leq 0 \\ & w_{23} - x_2 \leq 0 \\ & w_{32} - x_3 \leq 0 \\ & w_{45} - x_4 \leq 0 \\ & w_{46} - x_4 \leq 0 \\ & w_{51} - x_5 \leq 0 \\ & w_{54} - x_5 \leq 0 \\ & w_{56} - x_5 \leq 0 \\ & w_{64} - x_6 \leq 0 \\ & w_{65} - x_6 \leq 0 \\ & w_{12} - x_2 \leq 0 \\ & w_{15} - x_5 \leq 0 \\ & w_{21} - x_1 \leq 0 \\ & w_{23} - x_3 \leq 0 \\ & w_{32} - x_2 \leq 0 \\ & w_{45} - x_5 \leq 0 \\ & w_{46} - x_6 \leq 0 \\ & w_{51} - x_1 \leq 0 \\ & w_{54} - x_4 \leq 0 \\ & w_{56} - x_6 \leq 0 \\ & w_{64} - x_4 \leq 0 \\ & w_{65} - x_5 \leq 0 \\ & x_1 + x_3 - w_{13} \leq 1 \\ & x_3 + x_1 - w_{31} \leq 1 \\ & x_3 + x_4 - w_{34} \leq 1 \\ & x_4 + x_3 - w_{43} \leq 1 \\ & w_{13} \geq 0 \\ & w_{31} \geq 0 \\ & w_{34} \geq 0 \\ & w_{43} \geq 0 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\} \end{aligned}$$

Figura A.24 - Formulação linear do PQ considerando toda a matriz Q (PQLT).

Considerando o particionamento do problema em 2 *clusters* ($n = 2$), como apresentado na Figura A.25, o modelo $PQLT^n$ é dado na Figura A.26.

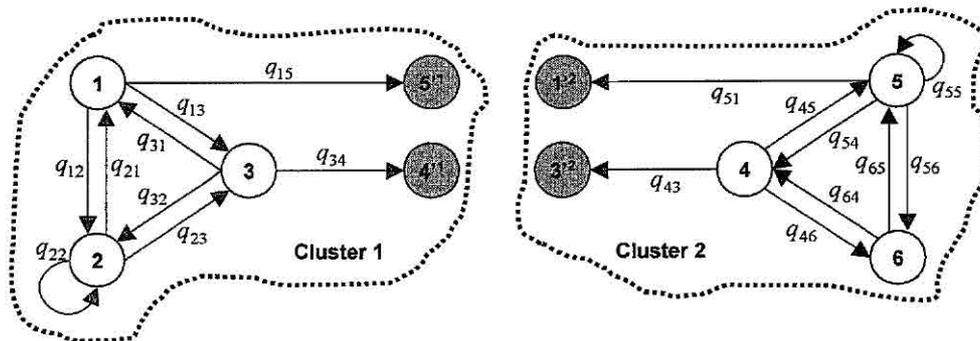


Figura A.25 - Particionamento do grafo na 3ª abordagem da DecLag.

No grafo acima, os vértices em cinza são as cópias necessárias para implementação da DecLag. Note que nesse caso, os vértices pertencentes às arestas relaxadas são copiados para o *cluster* oposto.

Relaxando as restrições de cópia (ver Figura A.26) no sentido lagrangiano, tem-se o modelo $DC_{\alpha\beta}PQLT^2$, apresentado na Figura A.27. Logo, os subproblemas gerados são apresentados na Figura A.28.

$PQLT^2$: $v(PQLT^2) = \text{Maximizar:}$

{ Cluster 1 }
 $-72 x_2 + 39 w_{12} - 99 w_{13} + 39 w_{21} + 78 w_{23} - 99 w_{31} + 78 w_{32}$
 $+ 33 w_{15'} - 28 w_{34'}$

{ Cluster 2 }
 $+ 30 x_5 + 84 w_{45} + 51 w_{46} + 84 w_{54} + 86 w_{56} + 51 w_{64} + 86 w_{65}$
 $- 28 w_{43'} + 33 w_{51'}$

Sujeito a:

$w_{12} - x_1 \leq 0$
 $w_{21} - x_2 \leq 0$
 $w_{23} - x_2 \leq 0$
 $w_{32} - x_3 \leq 0$
 $w_{12} - x_2 \leq 0$
 $w_{21} - x_1 \leq 0$
 $w_{23} - x_3 \leq 0$
 $w_{32} - x_2 \leq 0$
 $x_1 + x_3 - w_{13} \leq 1$
 $x_3 + x_1 - w_{31} \leq 1$
 $w_{13} \geq 0$
 $w_{31} \geq 0$

Cluster 1

$w_{45} - x_4 \leq 0$
 $w_{46} - x_4 \leq 0$
 $w_{54} - x_5 \leq 0$
 $w_{56} - x_5 \leq 0$
 $w_{64} - x_6 \leq 0$
 $w_{65} - x_6 \leq 0$
 $w_{45} - x_5 \leq 0$
 $w_{46} - x_6 \leq 0$
 $w_{54} - x_4 \leq 0$
 $w_{56} - x_6 \leq 0$
 $w_{64} - x_4 \leq 0$
 $w_{65} - x_5 \leq 0$

Cluster 2

$w_{15'} - x_1 \leq 0$
 $w_{15'} - x_{15'}^1 \leq 0$
 $x_3 + x_{4'}^1 - w_{34'} \leq 1$
 $w_{34'} \geq 0$

Inter-clusters: 1 \rightarrow 2

$w_{51'} - x_5 \leq 0$
 $w_{51'} - x_{1'}^2 \leq 0$
 $x_4 + x_{3'}^2 - w_{43'} \leq 1$
 $w_{43'} \geq 0$

Inter-clusters: 2 \rightarrow 1

$x_1 = x_{1'}^2$
 $x_3 = x_{3'}^2$
 $x_4 = x_{4'}^1$
 $x_5 = x_{5'}^1$
 $w_{15'} = w_{51'}$
 $w_{34'} = w_{43'}$

Restrições de cópia

$x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$

Figura A.26 - Modelo $PQLT^2$ utilizado na 3ª abordagem da DecLag.

$DC_{\alpha\beta}PQLT^2$: $v(DC_{\alpha\beta}PQLT^2) = \text{Maximizar:}$

$$\begin{aligned}
 & -72 x_2 + 39 w_{12} - 99 w_{13} + 39 w_{21} + 78 w_{23} - 99 w_{31} + 78 w_{32} + 33 w_{15'} \\
 & - 28 w_{34'} + 30 x_5 + 84 w_{45} + 51 w_{46} + 84 w_{54} + 86 w_{56} + 51 w_{64} + 86 w_{65} \\
 & - 28 w_{43'} + 33 w_{51'} - \alpha^1_4(x_4 - x^1_{4'}) - \alpha^1_5(x_5 - x^1_{5'}) - \alpha^2_1(x_1 - x^2_{1'}) \\
 & - \alpha^2_3(x_3 - x^2_{3'}) - \beta_{15}(w_{15'} - w_{51'}) - \beta_{34}(w_{34'} - w_{43'})
 \end{aligned}$$

Sujeito a:

$$\begin{aligned}
 & w_{12} - x_1 \leq 0 \\
 & w_{21} - x_2 \leq 0 \\
 & w_{23} - x_2 \leq 0 \\
 & w_{32} - x_3 \leq 0 \\
 & w_{12} - x_2 \leq 0 \\
 & w_{21} - x_1 \leq 0 \\
 & w_{23} - x_3 \leq 0 \\
 & w_{32} - x_2 \leq 0 \\
 & x_1 + x_3 - w_{13} \leq 1 \\
 & x_3 + x_1 - w_{31} \leq 1 \\
 & w_{13} \geq 0 \\
 & w_{31} \geq 0 \\
 & w_{45} - x_4 \leq 0 \\
 & w_{46} - x_4 \leq 0 \\
 & w_{54} - x_5 \leq 0 \\
 & w_{56} - x_5 \leq 0 \\
 & w_{64} - x_6 \leq 0 \\
 & w_{65} - x_6 \leq 0 \\
 & w_{45} - x_5 \leq 0 \\
 & w_{46} - x_6 \leq 0 \\
 & w_{54} - x_4 \leq 0 \\
 & w_{56} - x_6 \leq 0 \\
 & w_{64} - x_4 \leq 0 \\
 & w_{65} - x_5 \leq 0 \\
 & w_{15'} - x_1 \leq 0 \\
 & w_{15'} - x^1_{5'} \leq 0 \\
 & x_3 + x^1_{4'} - w_{34'} \leq 1 \\
 & w_{51'} - x_5 \leq 0 \\
 & w_{51'} - x^2_{1'} \leq 0 \\
 & x_4 + x^2_{3'} - w_{43'} \leq 1 \\
 & w_{34'}, w_{43'} \in \{0,1\} \\
 & x^1_{4'}, x^1_{5'}, x^2_{1'}, x^2_{3'} \in \{0,1\} \\
 & x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}
 \end{aligned}$$

Figura A.27 - Modelo $DC_{\alpha\beta}PQLT^2$ gerado na 3ª abordagem da DecLag.

<p>$DC_{\alpha\beta}PQLT_1: v(DC_{\alpha\beta}PQLT_1) =$</p> <p>Maximizar:</p> $-\alpha^2_1 x_1 - \alpha^2_3 x_3 - 72 x_2 + \alpha^1_4 x^1_4 + \alpha^1_5 x^1_5 + 39 w_{12} - 99 w_{13} + 39 w_{21} + 78 w_{23} - 99 w_{31} + 78 w_{32} + (33 - \beta_{15}) w_{15} + (-28 - \beta_{34}) w_{34}$ <p>Sujeito a:</p> $w_{12} - x_1 \leq 0$ $w_{21} - x_2 \leq 0$ $w_{23} - x_2 \leq 0$ $w_{32} - x_3 \leq 0$ $w_{12} - x_2 \leq 0$ $w_{21} - x_1 \leq 0$ $w_{23} - x_3 \leq 0$ $w_{32} - x_2 \leq 0$ $x_1 + x_3 - w_{13} \leq 1$ $x_3 + x_1 - w_{31} \leq 1$ $w_{13} \geq 0$ $w_{31} \geq 0$ $w_{15} - x_1 \leq 0$ $w_{15} - x^1_5 \leq 0$ $x_3 + x^1_4 - w_{34} \leq 1$ $w_{34} \in \{0,1\}$ $x^1_4, x^1_5 \in \{0,1\}$ $x_1, x_2, x_3 \in \{0,1\}$	<p>$DC_{\alpha\beta}PQLT_2: v(DC_{\alpha\beta}PQLT_2) =$</p> <p>Maximizar:</p> $-\alpha^1_4 x_4 + (30 - \alpha^1_5) x_5 + \alpha^2_1 x^2_1 + \alpha^2_3 x^2_3 + 84 w_{45} + 51 w_{46} + 84 w_{54} + 86 w_{56} + 51 w_{64} + 86 w_{65} + (-28 + \beta_{34}) w_{43} + (33 + \beta_{15}) w_{51}$ <p>Sujeito a:</p> $w_{45} - x_4 \leq 0$ $w_{46} - x_4 \leq 0$ $w_{54} - x_5 \leq 0$ $w_{56} - x_5 \leq 0$ $w_{64} - x_6 \leq 0$ $w_{65} - x_6 \leq 0$ $w_{45} - x_5 \leq 0$ $w_{46} - x_6 \leq 0$ $w_{54} - x_4 \leq 0$ $w_{56} - x_6 \leq 0$ $w_{64} - x_4 \leq 0$ $w_{65} - x_5 \leq 0$ $w_{51} - x_5 \leq 0$ $w_{51} - x^2_1 \leq 0$ $x_4 + x^2_3 - w_{43} \leq 1$ $w_{43} \in \{0,1\}$ $x^2_1, x^2_3 \in \{0,1\}$ $x_4, x_5, x_6 \in \{0,1\}$
--	--

Figura A.28 - Subproblemas gerados na 3ª abordagem da Declag.

A.4 Geração de Colunas

A.4.1 GC aplicada à LagClus

Considerando o mesmo particionamento apresentado na Figura A.10, define-se os seguintes vetores:

x^1	$\begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix}$	x^2	$\begin{matrix} x_4 \\ x_5 \\ x_6 \end{matrix}$	q^1_x	$\begin{matrix} 0 \\ -72 \\ 0 \end{matrix}$	q^2_x	$\begin{matrix} 0 \\ 30 \\ 0 \end{matrix}$
w^1	$\begin{matrix} w_{12} \\ w_{13} \\ w_{15} \\ w_{23} \\ w_{34} \end{matrix}$	w^2	$\begin{matrix} w_{45} \\ w_{46} \\ w_{56} \end{matrix}$	q^1_w	$\begin{matrix} 78 \\ -198 \\ 66 \\ 156 \\ -56 \end{matrix}$	q^2_w	$\begin{matrix} 168 \\ 102 \\ 172 \end{matrix}$

Figura A.29 - Vetores utilizados na GC com a LagClus.

A matriz A dos coeficientes das restrições relaxadas, em função das matrizes A_k , $k = 1, \dots, 2$, é apresentada na Figura A.30. Já a matriz D , dos coeficientes das restrições *intra-cluster*, é apresentada na Figura A.31.

$A_1 - \text{Cluster 1}$									$A_2 - \text{Cluster 2}$						
x_1	x_2	x_3	w_{12}	w_{13}	w_{15}	w_{23}	w_{34}	x_4	x_5	x_6	w_{45}	w_{46}	w_{56}		
0	0	0	0	0	1	0	0	0	-1	0	0	0	0		
0	0	1	0	0	0	0	-1	1	0	0	0	0	0		

Figura A.30 - Matriz referente às restrições relaxadas na GC com a LagClus.

$D_1 - \text{Cluster 1}$								$D_2 - \text{Cluster 2}$					
x_1	x_2	x_3	w_{12}	w_{13}	w_{15}	w_{23}	w_{34}	x_4	x_5	x_6	w_{45}	w_{46}	w_{56}
-1	0	0	1	0	0	0	0	0	0	0	0	0	0
0	-1	0	0	0	0	1	0	0	0	0	0	0	0
0	0	-1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	1	0	-1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0
-1	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	-1	0	0	1	0	0
0	0	0	0	0	0	0	0	-1	0	0	0	1	0
0	0	0	0	0	0	0	0	0	-1	0	0	0	1
0	0	0	0	0	0	0	0	0	-1	0	1	0	0
0	0	0	0	0	0	0	0	0	0	-1	0	1	0
0	0	0	0	0	0	0	0	0	0	-1	0	0	1

Figura A.31 - Matriz referente às restrições *intra-cluster* na GC com a LagClus.

Logo, o problema PQL^2 (ver Subseção 5.7.1) é descrito matricialmente como apresentado na Figura A.32. Conseqüentemente, os subproblemas Z_1^{LC} e Z_2^{LC} (ver Equação 5.115) são descritos nas Figuras A.33 e A.34, respectivamente.

$PQL^2: v(PQL^2) = \text{Maximizar:}$

$$\begin{pmatrix} 0 \\ -72 \\ 0 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 78 \\ -198 \\ 66 \\ 156 \\ -56 \end{pmatrix} \times \begin{pmatrix} w_{12} \\ w_{13} \\ w_{15} \\ w_{23} \\ w_{34} \end{pmatrix} + \begin{pmatrix} 0 \\ 30 \\ 0 \end{pmatrix} \times \begin{pmatrix} x_4 \\ x_5 \\ x_6 \end{pmatrix} + \begin{pmatrix} 168 \\ 102 \\ 172 \end{pmatrix} \times \begin{pmatrix} w_{45} \\ w_{46} \\ w_{56} \end{pmatrix}$$

Sujeito a:

$$\begin{array}{c} \begin{array}{cc} & \begin{array}{c} A_1 \end{array} \\ \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array} & \begin{array}{c} A_2 \\ \begin{array}{cccccccc} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \end{array} \end{array} \\ \times \begin{array}{c} \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ w_{12} \\ w_{13} \\ w_{15} \\ w_{23} \\ w_{34} \\ x_4 \\ x_5 \\ x_6 \\ w_{45} \\ w_{46} \\ w_{56} \end{array} \\ \begin{array}{c} \leq \\ \leq \end{array} \end{array} \begin{array}{c} \begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \end{array} \\ \end{array} \\ \mathbf{D}_1 \end{array}$$

Figura A.32 - Representação matricial de PQL^2 na GC com a LagClus.

$Z_1^{LC} = \text{Maximizar:}$

$$\begin{pmatrix} 0 \\ -72 \\ 0 \end{pmatrix} - \pi_1 \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} - \pi_2 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 78 \\ -198 \\ 66 \\ 156 \\ -56 \end{pmatrix} - \pi_1 \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} - \pi_2 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{pmatrix} \times \begin{pmatrix} w_{12} \\ w_{13} \\ w_{15} \\ w_{23} \\ w_{34} \end{pmatrix}$$

$$\begin{array}{c} \text{Sujeito a:} \\ \begin{array}{cccccccc} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \times \begin{array}{c} \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ w_{12} \\ w_{13} \\ w_{15} \\ w_{23} \\ w_{34} \end{array} \\ \begin{array}{c} \leq \\ \leq \\ \leq \\ \leq \\ \leq \\ \geq \\ \leq \\ \leq \end{array} \end{array} \begin{array}{c} \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} \end{array} \end{array}$$

Figura A.33 - Subproblema Z_1^{LC} na GC com a LagClus.

Z_2^{LC} = Maximizar:

$$\begin{pmatrix} 0 \\ 30 \\ 0 \end{pmatrix} -\pi_1 \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} -\pi_2 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_5 \\ x_6 \end{pmatrix} + \begin{pmatrix} 168 \\ 102 \\ 172 \end{pmatrix} -\pi_1 \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} -\pi_2 \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} w_{45} \\ w_{46} \\ w_{56} \end{pmatrix}$$

Sujeito a:

$$\begin{pmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_5 \\ x_6 \\ w_{45} \\ w_{46} \\ w_{56} \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Figura A.34 - Subproblema Z_2^{LC} na GC com a LagClus.

As variáveis duais π_1 e π_2 , apresentadas nos modelos Z_1^{LC} e Z_2^{LC} , são referentes à primeira e segunda restrições do PMR (ver Figura A.35). Para efeito de ilustração da formação do PMR inicial na Decomposição Dantzig-Wolfe (ver Seção 2.3), são consideradas duas soluções “triviais” para o problema:

Solução 1:

$$x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = 0$$

$$w_{12} = w_{13} = w_{15} = w_{23} = w_{34} = w_{45} = w_{46} = w_{56} = 0$$

Solução 2:

$$x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = 1$$

$$w_{12} = w_{13} = w_{15} = w_{23} = w_{34} = w_{45} = w_{46} = w_{56} = 1$$

Note que essas soluções são viáveis para o modelo PQL, e obviamente para os subproblemas (ver Figuras A.33 e A.34). Assim, essas soluções poderão formar um PMR inicial a partir da formação de colunas baseadas nos *clusters* gerados. Logo, tem-se:

Solução 1:

$$\mathbf{x}^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{x}^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{w}^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{w}^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Coluna 1:

$$\mathbf{q}_x^{1,1}\mathbf{x}^{1,1} + \mathbf{q}_w^{1,1}\mathbf{w}^{1,1} = \begin{bmatrix} 0 & -72 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 78 & -198 & 66 & 156 & -56 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \mathbf{0}$$

Coluna 2:

$$\mathbf{q}_x^{2,1}\mathbf{x}^{2,1} + \mathbf{q}_w^{2,1}\mathbf{w}^{2,1} = \begin{bmatrix} 0 & 30 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 168 & 102 & 172 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \mathbf{0}$$

Solução 2:

$$\mathbf{x}^1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{x}^2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{w}^1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{w}^2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Coluna 3:

$$\mathbf{q}_x^{1,2}\mathbf{x}^{1,2} + \mathbf{q}_w^{1,2}\mathbf{w}^{1,2} = \begin{bmatrix} 1 & -72 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 78 & -198 & 66 & 156 & -56 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = -\mathbf{26}$$

Coluna 4:

$$\mathbf{q}_x^{2,2}\mathbf{x}^{2,2} + \mathbf{q}_w^{2,2}\mathbf{w}^{2,2} = \begin{bmatrix} 1 & 30 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 168 & 102 & 172 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \mathbf{472}$$

Obtidos os custos dessas colunas, as restrições do PMR são montadas de acordo com as restrições 5.112 a 5.114. Logo, o PMR inicial apresentado na Figura A.35.

$\text{PMR}_{\text{PL}}^{\text{LC}}: v(\text{PMR}_{\text{PL}}^{\text{LC}}) = \text{Maximizar:}$ $0 \lambda_{11} + 0 \lambda_{21} - 26 \lambda_{12} + 472 \lambda_{22}$ <p style="text-align: center;">Sujeito a:</p> $0 \lambda_{11} + 0 \lambda_{21} + 1 \lambda_{12} - 1 \lambda_{22} \leq 0$ $0 \lambda_{11} + 0 \lambda_{21} + 0 \lambda_{12} + 1 \lambda_{22} \leq 1$ $\lambda_{11} + \lambda_{12} = 1$ $\lambda_{21} + \lambda_{22} = 1$ $\lambda_{11}, \lambda_{21}, \lambda_{12}, \lambda_{22} \geq 0$

Figura A.35 - Exemplo de PMR inicial para a GC com a LagClus.

Esse problema é resolvido (via CPLEX, por exemplo), e os vetores de variáveis duais $\vec{\pi}$ e $\vec{\mu}$ são obtidos. Na sequência, a função objetivo dos subproblemas (Figuras A.33 e A.34) é atualizada com os novos valores do vetor dual $\vec{\pi}$ e os subproblemas são resolvidos novamente (também via CPLEX, por exemplo) gerando assim novas colunas. Feito isso, é calculado o custo reduzido θ para cada nova coluna (soluções dos subproblemas) gerada ($\theta_1 = Z_1^{\text{LC}} - \mu_1$ e $\theta_2 = Z_2^{\text{LC}} - \mu_2$). As variáveis duais μ_1 e μ_2 são associadas às duas últimas restrições do PMR (restrições de convexidade). Caso os custos reduzidos dessas novas colunas sejam positivos, estas são inseridas no PMR. O processo é repetido até que um critério de parada seja satisfeito. Finalmente, as variáveis de decisão do PMR (λ) são integralizadas, ou seja, o PMR é resolvido de forma inteira, gerando assim uma solução para o problema PQ.

A.4.2 GC aplicada à DecLag

Considerando o particionamento apresentado na Figura A.25, define-se os vetores na Figura A.36.

x^1	x_1 x_2 x_3	x^2	x_4 x_5 x_6	q^1_x	0 -72 0	q^2_x	0 30 0
x'^1	$x'_{4'}$ $x'_{5'}$	x'^2	$x'^2_{1'}$ $x'^2_{3'}$	$q^1_{x'}$	0 0	$q^2_{x'}$	0 0
w^1	w_{12} w_{13} w_{21} w_{23} w_{31} w_{32}	w^2	w_{45} w_{46} w_{54} w_{56} w_{64} w_{65}	q^1_w	39 -99 39 78 -99 78	q^2_w	84 51 84 86 51 86
w'^1	$w'_{15'}$ $w'_{34'}$	w'^2	$w'_{43'}$ $w'_{51'}$	$q^1_{w'}$	33 -28	$q^2_{w'}$	-28 33

Figura A.36 - Vetores utilizados na GC com a DecLag.

As matrizes A_k referentes às restrições de cópia (relaxadas) são apresentadas na Figura A.37, e as matrizes D_k são apresentadas na Figura A.38.

x_1	x_2	x_3	$x'_{4'}$	$x'_{5'}$	w_{12}	w_{13}	w_{21}	w_{23}	w_{31}	w_{32}	$w'_{15'}$	$w'_{34'}$
1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	-1	0	0	0	0	0	0	0	0	0
0	0	0	0	-1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1

$A_1 =$

x_4	x_5	x_6	$x'^2_{1'}$	$x'^2_{3'}$	w_{45}	w_{46}	w_{54}	w_{56}	w_{64}	w_{65}	$w'_{43'}$	$w'_{51'}$
0	0	0	-1	0	0	0	0	0	0	0	0	0
0	0	0	0	-1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	-1
0	0	0	0	0	0	0	0	0	0	0	-1	0

$A_2 =$

Figura A.37 - Matrizes referentes às restrições relaxadas na GC com a DecLag.

$$D_1 =$$

x_1	x_2	x_3	$x_{4'}^1$	$x_{5'}^1$	w_{12}	w_{13}	w_{21}	w_{23}	w_{31}	w_{32}	$w_{15'}$	$w_{34'}$
-1	0	0	0	0	1	0	0	0	0	0	0	0
0	-1	0	0	0	0	0	1	0	0	0	0	0
0	-1	0	0	0	0	0	0	1	0	0	0	0
0	0	-1	0	0	0	0	0	0	0	1	0	0
0	-1	0	0	0	1	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	1	0	0	0	0	0
0	0	-1	0	0	0	0	0	1	0	0	0	0
0	-1	0	0	0	0	0	0	0	0	1	0	0
1	0	1	0	0	0	-1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	-1	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0
-1	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	-1	0	0	0	0	0	0	1	0
0	0	1	1	0	0	0	0	0	0	0	0	-1

$$D_2 =$$

x_4	x_5	x_6	$x_{1'}^2$	$x_{3'}^2$	w_{45}	w_{46}	w_{54}	w_{56}	w_{64}	w_{65}	$w_{43'}$	$w_{51'}$
-1	0	0	0	0	1	0	0	0	0	0	0	0
-1	0	0	0	0	0	1	0	0	0	0	0	0
0	-1	0	0	0	0	0	1	0	0	0	0	0
0	-1	0	0	0	0	0	0	1	0	0	0	0
0	0	-1	0	0	0	0	0	0	1	0	0	0
0	0	-1	0	0	0	0	0	0	0	1	0	0
0	-1	0	0	0	1	0	0	0	0	0	0	0
0	0	-1	0	0	0	1	0	0	0	0	0	0
-1	0	0	0	0	0	0	1	0	0	0	0	0
0	0	-1	0	0	0	0	0	1	0	0	0	0
-1	0	0	0	0	0	0	0	0	1	0	0	0
0	-1	0	0	0	0	0	0	0	0	1	0	0
0	-1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	-1	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	-1	0

Figura A.38 - Matrizes referentes às restrições *intra-cluster* na GC com a DecLag.

Logo, o problema $PQLT^2$ (ver Subseção 5.7.2) é descrito matricialmente como apresentado na Figura A.39. Consequentemente, os subproblemas Z_1^{DC} e Z_2^{DC} (ver Equação 5.125) são descritos nas Figuras A.40 e A.41, respectivamente.

$$\mathbf{x}^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{x}^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{x}'^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{x}'^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathbf{w}^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{w}^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{w}'^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{w}'^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Coluna 1:

$$\mathbf{q}_x^{1,1} \mathbf{x}^{1,1} + \mathbf{q}_{x'}^{1,1} \mathbf{x}'^{1,1} + \mathbf{q}_w^{1,1} \mathbf{w}^{1,1} + \mathbf{q}_{w'}^{1,1} \mathbf{w}'^{1,1} = 0$$

Coluna 2:

$$\mathbf{q}_x^{2,1} \mathbf{x}^{2,1} + \mathbf{q}_{x'}^{2,1} \mathbf{x}'^{2,1} + \mathbf{q}_w^{2,1} \mathbf{w}^{2,1} + \mathbf{q}_{w'}^{2,1} \mathbf{w}'^{2,1} = 0$$

Solução 2:

$$\mathbf{x}^1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{x}^2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{x}'^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{x}'^2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\mathbf{w}^1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{w}^2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{w}'^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{w}'^2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Coluna 3:

$$\begin{aligned} & \mathbf{q}_x^{1,2} \mathbf{x}^{1,2} + \mathbf{q}_{x'}^{1,2} \mathbf{x}'^{1,2} + \mathbf{q}_w^{1,2} \mathbf{w}^{1,2} + \mathbf{q}_{w'}^{1,2} \mathbf{w}'^{1,2} = \\ & \begin{bmatrix} 0 & -72 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{aligned}$$

$$+[39 \quad -99 \quad 39 \quad 78 \quad -99 \quad 78] \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + [33 \quad -28] \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} = -31$$

Coluna 4:

$$\begin{aligned} & \mathbf{q}_x^{2,2} \mathbf{x}^{2,2} + \mathbf{q}_{x'}^{2,2} \mathbf{x}'^{2,2} + \mathbf{q}_w^{2,2} \mathbf{w}^{2,2} + \mathbf{q}_{w'}^{2,2} \mathbf{w}'^{2,2} = \\ & [0 \quad 30 \quad 0] \times \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + [0 \quad 0] \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ & + [84 \quad 51 \quad 84 \quad 86 \quad 51 \quad 86] \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + [-28 \quad 33] \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 477 \end{aligned}$$

Obtidos os custos dessas colunas, as restrições do PMR são montadas de acordo com as restrições 5.122 a 5.124. Logo, o PMR inicial apresentado na Figura A.42. O restante do processo é análogo ao descrito na seção anterior.

<p>PMR^{DC}_{PL}: v(PMR^{DC}_{PL}) = Maximizar:</p> <p>$0 \lambda_{11} + 0 \lambda_{21} - 31 \lambda_{12} + 477 \lambda_{22}$</p> <p>Sujeito a:</p> <p>$0 \lambda_{11} + 0 \lambda_{21} + 1 \lambda_{12} - 1 \lambda_{22} = 0$</p> <p>$0 \lambda_{11} + 0 \lambda_{21} + 1 \lambda_{12} - 1 \lambda_{22} = 0$</p> <p>$0 \lambda_{11} + 0 \lambda_{21} - 1 \lambda_{12} + 1 \lambda_{22} = 0$</p> <p>$0 \lambda_{11} + 0 \lambda_{21} - 1 \lambda_{12} + 1 \lambda_{22} = 0$</p> <p>$0 \lambda_{11} + 0 \lambda_{21} + 1 \lambda_{12} - 1 \lambda_{22} = 0$</p> <p>$0 \lambda_{11} + 0 \lambda_{21} + 1 \lambda_{12} - 1 \lambda_{22} = 0$</p> <p>$\lambda_{11} + \lambda_{12} = 1$</p> <p>$\lambda_{21} + \lambda_{22} = 1$</p> <p>$\lambda_{11}, \lambda_{21}, \lambda_{12}, \lambda_{22} \geq 0$</p>
--

Figura A.42 - Exemplo de PMR inicial para a GC com a DecLag.

A.5 Formulação posiforme

No modelo apresentado na Figura A.4, os termos quadráticos com coeficientes negativos, $2q_{ij}x_i x_j$, $\forall q_{ij} < 0$, são substituídos pelos termos $2q_{ij} + |2q_{ij}|\bar{x}_i + |2q_{ij}|x_i\bar{x}_j$, e os termos $q_{ii}x_i$, $\forall q_{ii} < 0$, pelos termos $q_{ii} + |q_{ii}|\bar{x}_i$. Logo, o modelo baseado na posiforme para o PQ (PPQ) é apresentado na Figura A.43.

PPQ: $v(PPQ) = \text{Maximizar:}$

$$-72 + 72\bar{x}_2 + 30x_5 + 78x_1x_2 - 198 + 198\bar{x}_1 + 198x_1\bar{x}_3 + 66x_1x_5 + 156x_2x_3 \\ - 56 + 56\bar{x}_3 + 56x_3\bar{x}_4 + 168x_4x_5 + 102x_4x_6 + 172x_5x_6$$

Sujeito a:

$$x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$$

Figura A.43 - Formulação baseada na posiforme (PPQ).

A partir dessa formulação, é criado um grafo cujas arestas são formadas entre as variáveis e seus complementos (Figura A.44).

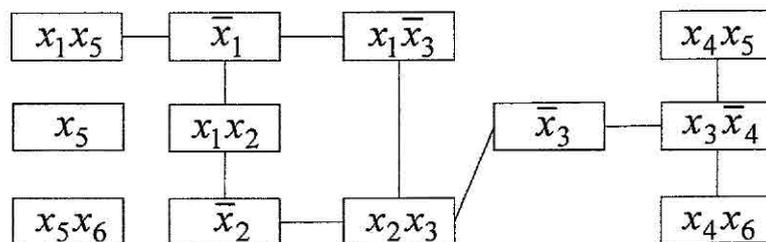


Figura A.44 - Grafo baseado nos termos da função objetivo do PPQ.

Cada vértice desse grafo é renomeado de acordo com uma variável y_i . Logo, esse grafo é reescrito como apresentado na Figura A.45.

Como descrito na Seção 5.8, a partir desse grafo o modelo PPQ é linearizado (PPQL) com a utilização das arestas para montar as restrições. Assim, PPQL é apresentado na Figura A.46.

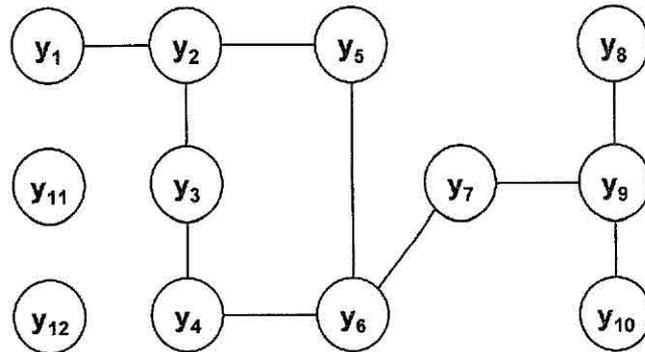


Figura A.45 - Grafo para o PPQ baseado nas variáveis y_i .

PPQL: $v(PPQL) = \text{Maximizar:}$

$$66 y_1 + 198 y_2 + 78 y_3 + 72 y_4 + 198 y_5 + 156 y_6 + 56 y_7 + 168 y_8 + 56 y_9 + 102 y_{10} + 30 y_{11} + 172 y_{12} - 72 - 198 - 56$$

Sujeito a:

$$y_1 + y_2 \leq 1$$

$$y_2 + y_3 \leq 1$$

$$y_2 + y_5 \leq 1$$

$$y_3 + y_4 \leq 1$$

$$y_4 + y_6 \leq 1$$

$$y_5 + y_6 \leq 1$$

$$y_6 + y_7 \leq 1$$

$$y_7 + y_9 \leq 1$$

$$y_8 + y_9 \leq 1$$

$$y_9 + y_{10} \leq 1$$

$$y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}, y_{12} \in \{0, 1\}$$

Figura A.46 - Formulação linear do PPQ (PPQL).

Como pode ser observado na Figura A.45, os vértices y_{11} e y_{12} estão desconectados dos demais vértices, e conseqüentemente poderão ser eliminados do grafo (e do problema), e seus coeficientes adicionados à função objetivo. Logo, o modelo PPQL pode ser reescrito como apresentado na Figura A.47, e o respectivo grafo na Figura A.48.

PPQL: $v(PPQL) = \text{Maximizar:}$

$$66 y_1 + 198 y_2 + 78 y_3 + 72 y_4 + 198 y_5 + 156 y_6 + 56 y_7 + 168 y_8 + 56 y_9 + 102 y_{10} + 30 + 172 - 72 - 198 - 56$$

Sujeito a:

$$y_1 + y_2 \leq 1$$

$$y_2 + y_3 \leq 1$$

$$y_2 + y_5 \leq 1$$

$$y_3 + y_4 \leq 1$$

$$y_4 + y_6 \leq 1$$

$$y_5 + y_6 \leq 1$$

$$y_6 + y_7 \leq 1$$

$$y_7 + y_9 \leq 1$$

$$y_8 + y_9 \leq 1$$

$$y_9 + y_{10} \leq 1$$

$$y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10} \in \{0,1\}$$

Figura A.47 - Formulação do PPQL com a eliminação de vértices desconectados.

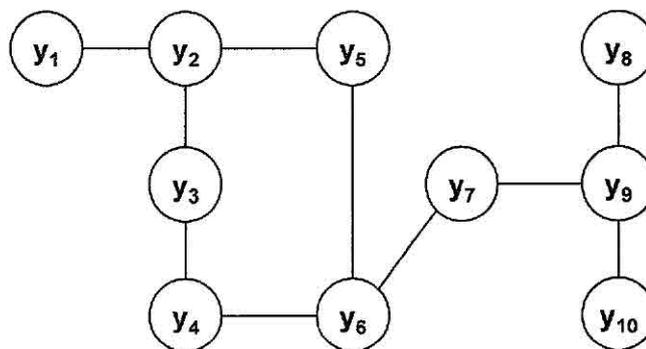


Figura A.48 - Grafo para a formulação do PPQL sem os vértices desconectados.

A.5.1 LagClus aplicada ao PPQL

Considerando o particionamento do problema em 2 *clusters* ($n = 2$), como apresentado na Figura A.49, o modelo $PPQL^n$ é dado na Figura A.50.

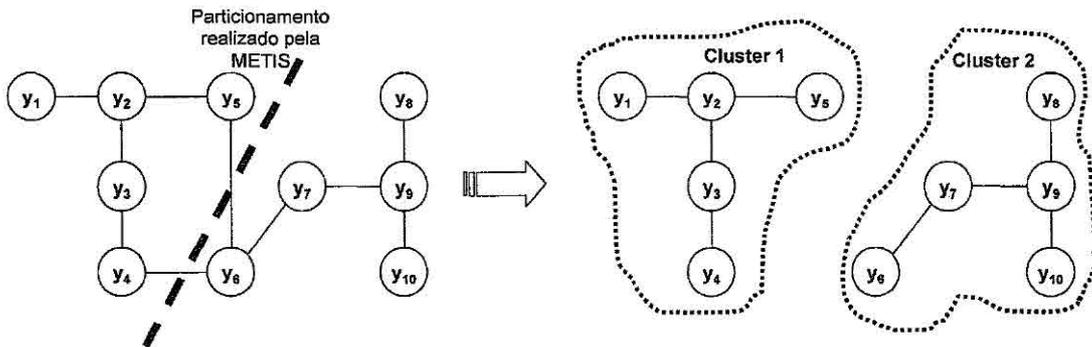


Figura A.49 - Particionamento do grafo baseado na posiforme com a LagClus.

$$\begin{aligned}
 &PPQL^2: v(PPQL^2) = \text{Maximizar:} \\
 &66 y_1 + 198 y_2 + 78 y_3 + 72 y_4 + 198 y_5 \quad \{ \text{Cluster 1} \} \\
 &+ 156 y_6 + 56 y_7 + 168 y_8 + 56 y_9 + 102 y_{10} \quad \{ \text{Cluster 2} \} \\
 &- 124 \\
 &\text{Sujeito a:} \\
 &\left. \begin{aligned} y_1 + y_2 &\leq 1 \\ y_2 + y_3 &\leq 1 \\ y_2 + y_5 &\leq 1 \\ y_3 + y_4 &\leq 1 \end{aligned} \right\} \text{Cluster 1} \\
 &\left. \begin{aligned} y_6 + y_7 &\leq 1 \\ y_7 + y_9 &\leq 1 \\ y_8 + y_9 &\leq 1 \\ y_9 + y_{10} &\leq 1 \end{aligned} \right\} \text{Cluster 2} \\
 &\left. \begin{aligned} y_4 + y_6 &\leq 1 \\ y_5 + y_6 &\leq 1 \end{aligned} \right\} \text{Inter-Clusters} \\
 &y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10} \in \{0,1\}
 \end{aligned}$$

Figura A.50 - Modelo $PPQL^2$ utilizado na LagClus.

Relaxando as restrições *inter-clusters* (ver Figura A.50) no sentido lagrangiano, tem-se o modelo $LC_{\alpha}PPQL^2$, apresentado na Figura A.51. Esse modelo pode ser dividido em dois ($n = 2$) subproblemas independentes, como apresentado na Figura A.52.

$LC_{\alpha}PPQL^2: v(LC_{\alpha}PPQL^2) = \text{Maximizar:}$ $66 y_1 + 198 y_2 + 78 y_3 + 72 y_4 + 198 y_5 + 156 y_6 + 56 y_7 + 168 y_8 + 56 y_9 + 102 y_{10} - \alpha_{46} (y_4 + y_6 - 1) - \alpha_{56} (y_5 + y_6 - 1)$ <p>Sujeito a:</p> $y_1 + y_2 \leq 1$ $y_2 + y_3 \leq 1$ $y_2 + y_5 \leq 1$ $y_3 + y_4 \leq 1$ $y_6 + y_7 \leq 1$ $y_7 + y_9 \leq 1$ $y_8 + y_9 \leq 1$ $y_9 + y_{10} \leq 1$ $y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10} \in \{0,1\}$

Figura A.51 - Modelo $LC_{\alpha}PPQL^2$ gerado na LagClus.

$LC_{\alpha}PPQL_1: v(LC_{\alpha}PPQL_1) = \text{Maximizar:}$ $66 y_1 + 198 y_2 + 78 y_3 + (72 - \alpha_{46}) y_4 + (198 - \alpha_{56}) y_5 +$ <p>Sujeito a:</p> $y_1 + y_2 \leq 1$ $y_2 + y_3 \leq 1$ $y_2 + y_5 \leq 1$ $y_3 + y_4 \leq 1$ $y_1, y_2, y_3, y_4, y_5 \in \{0,1\}$	$LC_{\alpha}PPQL_2: v(LC_{\alpha}PPQL_2) = \text{Maximizar:}$ $(156 - \alpha_{46} - \alpha_{56}) y_6 + 56 y_7 + 168 y_8 + 56 y_9 + 102 y_{10}$ <p>Sujeito a:</p> $y_6 + y_7 \leq 1$ $y_7 + y_9 \leq 1$ $y_8 + y_9 \leq 1$ $y_9 + y_{10} \leq 1$ $y_6, y_7, y_8, y_9, y_{10} \in \{0,1\}$
---	---

Figura A.52 - Subproblemas gerados para a posiforme na LagClus.

A.5.2 DecLag aplicada ao PPQL

Considerando o particionamento do problema em 2 *clusters* ($n = 2$) e as cópias de vértices necessárias, como apresentado na Figura A.53, o modelo $PPQL^n$ é dado na Figura A.54.

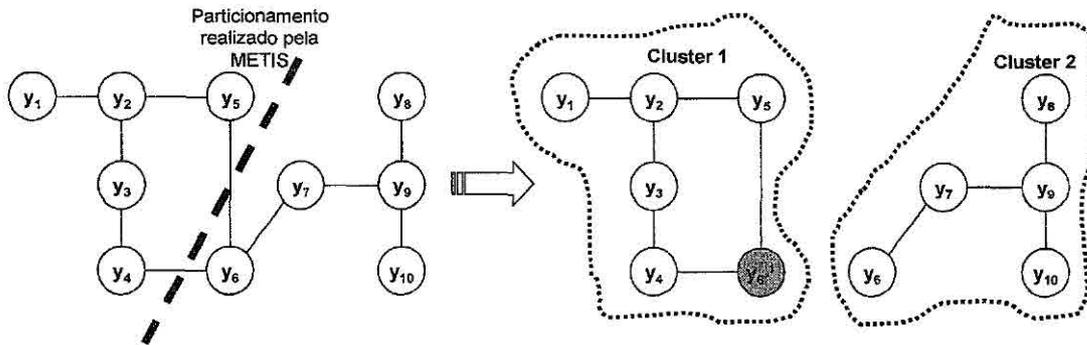


Figura A.53 - Particionamento do grafo baseado na posiforme com a DecLag.

$$\begin{aligned}
 &PPQL^2: v(PPQL^2) = \text{Maximizar:} \\
 &66 y_1 + 198 y_2 + 78 y_3 + 72 y_4 + 198 y_5 \quad \{ \text{Cluster 1} \} \\
 &+ 156 y_6 + 56 y_7 + 168 y_8 + 56 y_9 + 102 y_{10} \quad \{ \text{Cluster 2} \} \\
 &- 124 \\
 &\text{Sujeito a:} \\
 &\left. \begin{aligned} y_1 + y_2 &\leq 1 \\ y_2 + y_3 &\leq 1 \\ y_2 + y_5 &\leq 1 \\ y_3 + y_4 &\leq 1 \end{aligned} \right\} \text{Cluster 1} \\
 &\left. \begin{aligned} y_6 + y_7 &\leq 1 \\ y_7 + y_9 &\leq 1 \\ y_8 + y_9 &\leq 1 \\ y_9 + y_{10} &\leq 1 \end{aligned} \right\} \text{Cluster 2} \\
 &\left. \begin{aligned} y_4 + y_6^1 &\leq 1 \\ y_5 + y_6^1 &\leq 1 \end{aligned} \right\} \text{Inter-Clusters} \\
 &y_6 = y_6^1 \quad \} \text{Restrição de Cópia} \\
 &y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10} \in \{0, 1\}
 \end{aligned}$$

Figura A.54 - Modelo $PPQL^2$ utilizado na DecLag.

Relaxando as restrições de cópia (ver Figura A.54) no sentido lagrangiano, tem-se o modelo $DC_{\alpha}PPQL^2$, apresentado na Figura A.55. Esse modelo pode ser dividido em dois ($n = 2$) subproblemas independentes, como apresentado na Figura A.56.

<p>$DC_{\alpha}PPQL^2: v(DC_{\alpha}PPQL^2) = \text{Maximizar:}$</p> $66 y_1 + 198 y_2 + 78 y_3 + 72 y_4 + 198 y_5 + 156 y_6 + 56 y_7 + 168 y_8 + 56 y_9$ $+ 102 y_{10} - \alpha^1_6 (y_6 - y^1_6)$ <p>Sujeito a:</p> $y_1 + y_2 \leq 1$ $y_2 + y_3 \leq 1$ $y_2 + y_5 \leq 1$ $y_3 + y_4 \leq 1$ $y_6 + y_7 \leq 1$ $y_7 + y_9 \leq 1$ $y_8 + y_9 \leq 1$ $y_9 + y_{10} \leq 1$ $y_4 + y^1_6 \leq 1$ $y_5 + y^1_6 \leq 1$ $y^1_6 \in \{0,1\}$ $y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10} \in \{0,1\}$
--

Figura A.55 - Modelo $DC_{\alpha}PPQL^2$ gerado na DecLag.

<p>$DC_{\alpha}PPQL_1: v(DC_{\alpha}PPQL_1) = \text{Maximizar:}$</p> $66 y_1 + 198 y_2 + 78 y_3 + 72 y_4 + 198 y_5$ $+ \alpha^1_6 y^1_6$ <p>Sujeito a:</p> $y_1 + y_2 \leq 1$ $y_2 + y_3 \leq 1$ $y_2 + y_5 \leq 1$ $y_3 + y_4 \leq 1$ $y_4 + y^1_6 \leq 1$ $y_5 + y^1_6 \leq 1$ $y^1_6 \in \{0,1\}$ $y_1, y_2, y_3, y_4, y_5 \in \{0,1\}$	<p>$DC_{\alpha}PPQL_2: v(DC_{\alpha}PPQL_2) = \text{Maximizar:}$</p> $(156 - \alpha^1_6) y_6 + 56 y_7 + 168 y_8 + 56 y_9$ $+ 102 y_{10}$ <p>Sujeito a:</p> $y_6 + y_7 \leq 1$ $y_7 + y_9 \leq 1$ $y_8 + y_9 \leq 1$ $y_9 + y_{10} \leq 1$ $y_6, y_7, y_8, y_9, y_{10} \in \{0,1\}$
--	--

Figura A.56 - Subproblemas gerados para a posiforme na DecLag.

