



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

PABLO LUIZ BRAGA SOARES

PROBLEMAS QUADRÁTICOS BINÁRIOS: ABORDAGEM TEÓRICA E
COMPUTACIONAL

FORTALEZA
2018

PABLO LUIZ BRAGA SOARES

PROBLEMAS QUADRÁTICOS BINÁRIOS: ABORDAGEM TEÓRICA E
COMPUTACIONAL

Tese apresentada ao Programa de Pós-graduação em Ciência da Computação do Departamento de Computação da Universidade Federal do Ceará, como parte dos requisitos necessários para a obtenção do título de Doutor em Ciência da Computação. Área de concentração: Algoritmos e Otimização.

Orientador: Prof. Dr. Manoel Bezerra Campêlo Neto

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- S656p Soares, Pablo Luiz Braga.
Problemas quadráticos binários : abordagem teórica e computacional / Pablo Luiz Braga Soares. – 2018.
125 f. : il. color.
- Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em
Ciência da Computação, Fortaleza, 2018.
Orientação: Prof. Dr. Manoel Bezerra Campêlo Neto.
1. t-Linearização. 2. Suavização Hiperbólica. 3. Programação Quadrática 0-1. 4. Desigualdades Válidas. I.
Título.

CDD 005

PABLO LUIZ BRAGA SOARES

PROBLEMAS QUADRÁTICOS BINÁRIOS: ABORDAGEM TEÓRICA E
COMPUTACIONAL

Tese apresentada ao Programa de Pós-graduação em Ciência da Computação do Departamento de Computação da Universidade Federal do Ceará, como parte dos requisitos necessários para a obtenção do título de Doutor em Ciência da Computação. Área de concentração: Algoritmos e Otimização.

Aprovada em: __ / __ / 2018.

BANCA EXAMINADORA

Prof. Dr. Manoel Bezerra Campêlo Neto (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Rafael Castro de Andrade
Universidade Federal do Ceará (UFC)

Prof. Dr. Carlos Diego Rodrigues
Universidade Federal do Ceará (UFC)

Prof. Dr. Philippe Yves Paul Michelon
Université D'Avignon

Prof. Dr. Adilson Elias Xavier
Universidade Federal do Rio de Janeiro (UFRJ)

*Dedico esse trabalho a toda minha família,
em especial aos meus pais Francisco Soares e
Rita Braga.*

AGRADECIMENTOS

À Deus por está sempre cuidando e guiando minhas caminhadas, além de escutar e aconselhar meus pensamentos em todos os momentos da minha vida.

Ao meu pai, Francisco Soares (Ticão de Zé Raul), meu herói, meu amigo, meu confidente, simplesmente meu... que apesar de distante nunca deixou faltar nada para minha formação.

À minha mãe, Rita Braga, que assumiu o papel de mãe e pai ao mesmo tempo desde que passamos a morar em Mossoró. Pelo seu amor, carinho, apoio e confiança. A ela dedico todo o mérito da minha educação, formação e caráter.

À minha irmã, Larissa Braga, pelos cuidados e quando criança e pelas preocupações da juventude. Você é grande e vencerá.

Ao meu avô de Janduís, José Soares (Zé Raul), o homem mais bondoso e sábio que já conheci. Espero ter herdado essas suas características, além da capacidade de ver sempre o lado bom das coisas apesar de tudo.

À Francisca Soares (Neném Raul), minha avó de Janduís, que sempre me recebeu de braços abertos em todos os momentos.

Ao meu orientador, Prof. Manoel Bezerra Campêlo Neto, pela excelente pessoa, pesquisador brilhante, professor excepcional, a quem devo toda minha evolução acadêmica e conquista. Nunca deixou de acreditar que daria certo, mesmo eu já não acreditando mais que era possível. Você me orientou com maestria e me ajudou como nenhum outro. Obrigado meu mestre serei sempre grato pelos seus ensinamentos.

À minha grande amiga, Priscila Jácome, que sempre me dar os melhores conselhos amorosos mas que eu nunca sigo hahahah.

À todos meus amigos do grupo ParGo, pessoas maravilhosas que dividiram nesses anos os dias de anseio, dias de luta, viagens, sorrisos, jogos, bebedeiras/festas em especial os amigos Rennan Dantas e Rafael Teixeira que me acolheram em suas casas quando precisei.

À todos os meus alunos da UFC Campus Russas, em especial aos do grupo M.O.R.T.A.L representados aqui por: Carlos Victor Arr@mbado(meu primeiro aluno de TCC), Marcus Araújo(Grande amigo que encabeça minhas ideias e faz as coisas acontecerem), Cícero Marcelo(Meu bolsista que sempre segurou a peteca quando precisei ficar ausente) e Daniel Rebouças(Meu aluno de iniciação científica que contribuiu significativamente na parte final dessa tese).

Finalmente, a toda minha família (tias, tios, primos e primas), que de uma forma ou de outra estão sempre me apoiando e ajudando.

“Tudo seria fácil se não fossem as dificuldades.”

Barão de Itararé

RESUMO

O objetivo principal deste trabalho é mostrar a aplicação de métodos das áreas de programação linear e não linear para tratar problemas de otimização cuja formulação 0 – 1 natural tem uma função objetivo quadrática. Para esse fim, desenvolvemos a extensão da técnica da t -linearização e mostramos como aplicá-la a problemas quadráticos tais como Corte Máximo, Diversidade Máxima e Clique Máxima Ponderada em Aresta. Adicionalmente, revisamos a técnica da suavização hiperbólica e mostramos como aplicá-la ao problema do corte máximo. Realizamos um estudo teórico de cada problema, onde apresentamos particularmente o desenvolvimento de novas desigualdades válidas e heurísticas simples para cada um deles. Mostramos como gerar restrições t -linearizadas mais fortes para os problemas da diversidade máxima e clique máxima ponderada em aresta. Executamos experimentos computacionais para avaliar o desempenho de cada técnica mostrada em separado, assim como combinadas com as novas restrições propostas. Os experimentos atestam a qualidade dos limites obtidos bem como a força das novas restrições. Construímos um método exato para cada problema, baseado nas técnicas apresentadas. Para o corte máximo e clique máxima ponderada em aresta, comparamos nosso método com formulações linearizadas. Já para diversidade máxima, comparamos nosso método com formulações linearizadas e com uma implementação nossa do melhor método exato, que não utiliza resolvedor matemático, conhecido da literatura. Experimentos computacionais com instâncias da literatura mostram um desempenho superior do nosso método.

Palavras-chave: t -Linearização. Suavização Hiperbólica. Programação Quadrática 0 – 1. Desigualdades Válidas.

ABSTRACT

The main objective of this work is to show the application of linear and nonlinear programming methods to deal with problems whose natural 0–1 formulation has a quadratic objective function. To this end, we develop an extension of the t -linearization technique and show how to apply it to quadratic problems such as Maximum Cut, Maximum Diversity and Maximum Edge-Weighted Clique. In addition, we review the hyperbolic smoothing technique and show how to apply it to the maximum cut problem. We carry out a theoretical study of each problem, where we particularly present the development of new valid inequalities for each of them. We perform computational experiments to evaluate the performance of each technique separately, as well as combined with the proposed new constraints. The experiments attest the quality of the bounds obtained as well as the strength of the new inequalities. We develop an exact method for each problem, based on the presented techniques. For the maximum cut and maximum edge-weighted clique, we compare our method with linearized formulations. For maximum diversity, we compare our method with linearized formulations and with our implementation of the best exact method, which does not use mathematical solvers, known in the literature. Computational experiments with instances of the literature show superior performance of our method.

Keywords: t -Linearization. Hyperbolic Smoothing. 0–1 Quadratic Programming. New Valid Inequalities

LISTA DE FIGURAS

Figura 1 – Diferença entre $\Theta := \sum_i \sum_{j < i} c_{\pi(j)\pi(i)} \hat{x}_{\pi(i)}$ e $\Theta' := \sum_i \sum_{j < i} c_{\pi'(j)\pi'(i)} \hat{x}_{\pi'(i)}$. Em azul as posições diferentes entre π e π' . As setas indicam os coeficientes que aparecem em Θ mas não em Θ' (figura superior) e vice-versa (figura inferior)	24
Figura 2 – Coeficientes α que aparecem em $\Theta(\pi, N_0, N_1; x)$. As setas indicam a que variáveis o coeficiente $\alpha_{\pi(j)\pi(i)}$, $j < i$, está associado.	29
Figura 3 – Coeficientes β que aparecem em $\Theta(\pi, N_0, N_1; x)$. As setas indicam a que variáveis o coeficiente $\beta_{\pi(j)\pi(i)}$, $j < i$, está associado.	29
Figura 4 – Diferença entre $\Theta(\pi, N_0, N_1; \hat{x})$ e $\Theta(\pi', N_0, N_1; \hat{x})$ quando $\pi(l+1) \in N_1$. Em azul as posições diferentes entre π e π' . As setas indicam os coeficientes α (Fig. a) e β (Fig. b) que aparecem em $\Theta(\pi, N_0, N_1; \hat{x})$ mas não em $\Theta(\pi', N_0, N_1; \hat{x})$ (figura superior) e vice-versa (figura inferior), apresentando as variáveis associadas.	31
Figura 5 – Diferença entre $\Theta(\pi, N_0, N_1; \hat{x})$ e $\Theta(\pi', N_0, N_1; \hat{x})$ quando $\pi(l+1) \in N_0$. Em azul as posições diferentes entre π e π' . As setas indicam os coeficientes α (Fig. a) e β (Fig. b) que aparecem em $\Theta(\pi, N_0, N_1; \hat{x})$ mas não em $\Theta(\pi', N_0, N_1; \hat{x})$ (figura superior) e vice-versa (figura inferior), apresentando as variáveis associadas.	32
Figura 6 – $ u $ e $\theta(u, \gamma)$, para $\gamma = [1, 0.8, 0.5, 0.3]$	39
Figura 7 – $g(y_i)$ e $\phi(\lambda, \tau, y_i)$, para $\tau = [1, 0.8, 0.5, 0.3]$	40
Figura 8 – Fixação $x_k = 1$ em uma solução descendente de \hat{x} : limite inferior (a) maior que limite superior (b).	49
Figura 9 – Fixação de variáveis.	50
Figura 10 – Decisão heurística. Casos em que $\Delta(k^*, \hat{x}) > 0$, ou seja, $ C_{01}(k^*, \hat{x}) < C_2(k^*, \hat{x})$	65
Figura 11 – Gap médio com n fixo e variação de d	72
Figura 12 – Gap médio com d fixo e variação de n	73
Figura 13 – Ilustração das expressões de $c_{\min}(\bar{x}, v)$ e $c_{\max}(\bar{x}, v)$. Arestas pontilhadas compõem $c_{\min}(\bar{x}, v)$, arestas tracejadas compõem $c_{\max}(\bar{x}, v)$ e arestas cheias compõem ambos.	81
Figura 14 – Ilustração dos termos de $\delta_{uv}(\bar{x})$. Os sinais nas arestas indicam a contribuição de cada custo para a expressão.	82
Figura 15 – Gap médio obtido pelas relaxações $F_2, F_3, (F_1)_t^\pi, F_m$ e $(F_1)_t^\pi + RF$. . .	87
Figura 16 – Gap médio obtido pelas relaxações $F_2, F_3, (F_1)_t^\pi, F_m$ e $(F_1)_t^\pi + RF$. . .	88
Figura 17 – Fluxograma do método exato. $(BB)_t^\pi$ segue as setas verticais. $(F_1)_t^\pi$ segue o caminho alternativo.	89

Figura 18 – Número médio de restrições geradas por $(F_1)_t^\pi$ e $(BB)_t^\pi$	90
Figura 19 – Comparação do GAP médio obtido por FCA_m, FCA_2, FCA_3 e $(FCA_1)_t^\pi$	107
Figura 20 – Comparação do N_{rg} médio obtido por $(FCA_1)_t^\pi$ e FCA_3	108
Figura 21 – Gap médio(30 instâncias positivas) obtido pelas formulações $FCA_1,$ FCA_m, FCA_2, FCA_3 e $(BBCA)_t^\pi$	111
Figura 22 – Cpu médio(30 instâncias positivas) obtido pelas formulações $FCA_1,$ FCA_m, FCA_2, FCA_3 e $(BBCA)_t^\pi$	112
Figura 23 – Gap médio(30 instâncias mistas) obtido pelas formulações $FCA_1, FCA_m,$ FCA_2, FCA_3 e $(BBCA)_t^\pi$	113
Figura 24 – Cpu médio(30 instâncias mistas) obtido pelas formulações $FCA_1, FCA_m,$ FCA_2, FCA_3 e $(BBCA)_t^\pi$	113

LISTA DE TABELAS

Tabela 1 – Valores individuais de y_{ji}^*	37
Tabela 2 – Peso do corte da partição (S, \bar{S}) representado pelo vetor binário $x \in \{0, 1\}^n$	44
Tabela 3 – Peso do corte da (S, \bar{S}) representado pelo vetor binário $x \in \{-1, 1\}^n$. .	45
Tabela 4 – Equivalência entre $x_i + x_j - 2x_i x_j$, $ x_i - x_j $ e z_{ij}	59
Tabela 5 – CPU, gap , R_f e N_{rg} obtidos pela t -linearização, linearização clássica e linearização Glover	68
Tabela 6 – LB e CPU obtidos pelas heurísticas gulosas, SH , $SH + \Delta$ e $SH + \Delta + BL$	69
Tabela 7 – Comparação entre $(FL)_t$ e $(FL)_{tGRV}$	71
Tabela 8 – Comparação entre $(FL)_{tGRV}$ e $(FL)_{tLRV}$	72
Tabela 9 – Principais aplicações do problema da diversidade máxima	75
Tabela 10 – GAP%, CPU(s) e N_{rg} médio para as relaxações $(F_1)_t^\pi$ e $(F_1)_t^\pi + RF$ nas instâncias Silva	86
Tabela 11 – GAP%, CPU(s) e N_{rg} médio para as relaxações $(F_1)_t^\pi$ e $(F_1)_t^\pi + RF$ nas instâncias Glover2	86
Tabela 12 – GAP%, CPU(s), R_f e N_{rg} médio para as relaxações F_2 , F_3 , F_m e $(F_1)_t^\pi +$ RF nas instâncias Silva	87
Tabela 13 – GAP%, CPU(s), R_f e N_{rg} médio para as relaxações F_2 , F_3 , F_m e $(F_1)_t^\pi +$ RF nas instâncias Glover2	87
Tabela 14 – Valores médios de GAP, CPU e N_{rg} e #Opt obtidos por $(F_1)_t^\pi$ e $(BB)_t^\pi$	90
Tabela 15 – Resultados com as instâncias <i>Silva</i> obtidos pelas formulações F_1 e F_m , e os métodos <i>BBmax</i> e $(BB)_t^\pi$	92
Tabela 16 – Resultados com as instâncias <i>Glover2</i> obtidos pelas formulações F_1 e F_m , e os métodos <i>BBmax</i> e $(BB)_t^\pi$	93
Tabela 17 – Versões do PCMPA	95
Tabela 18 – GAP, R_f e N_{rg} obtidos pelas relaxações FCA_m , FCA_2 , FCA_3 e $(FCA_1)_t^\pi$ nas instâncias positivas	107
Tabela 19 – GAP, R_f e N_{rg} obtidos pelas relaxações FCA_m , FCA_2 , FCA_3 e $(FCA_1)_t^\pi$ nas instâncias mistas	108
Tabela 20 – GAP e CPU obtidos pelas formulações FCA_1 , FCA_m , FCA_2 , FCA_3 e $(BBCA)_t^\pi$ nas instâncias positivas com $m = \lfloor n/2 \rfloor$	111
Tabela 21 – GAP e CPU obtidos pelas formulações FCA_1 , FCA_m , FCA_2 , FCA_3 e $(BBCA)_t^\pi$ nas instâncias mistas com $m = \lfloor n/2 \rfloor$	112

LISTA DE ABREVIATURAS E SIGLAS

Biq Mac	Binary Quadratic and Max Cut Solver
DNA	Deoxyribonucleic Acid
GRASP	Greedy Randomized Adaptive Search Procedure
IBM	International Business Machines
IDE	Integrates Development Environment
ITS	Iterative Tabu Search
LS-TPG	Lagrangian Smoothing Truncated Project Gradient
LS-TFW	Lagrangian Smoothing Truncated Frank-Wolfe
PCMPA	Problema da Clique Máxima Ponderada em Aresta
PDM	Problema da Diversidade Máxima
PI	Programação Inteira
PIM	Programação Inteira Mista
PL	Programação Linear
PQ	Problema Quadrático
PQM	Problema Quadrático da Mochila
PR	Path-Relinking
PSD	Programação Semidefinida
RAM	Random Access Memory
SBPO	Simpósio Brasileiro de Pesquisa Operacional
SH	Suavização Hiperbólica
SS	Scatter-Search
UQP	Unconstrained Quadratic Programming
VNS	Variable Neighborhood Search
VSLI	Very Large Scale Integration
WCP	Weighted Maximal Clique

SUMÁRIO

1	INTRODUÇÃO	15
2	METODOLOGIA	19
2.1	Linearização Clássica	19
2.2	Linearização de Glover	20
2.3	t -Linearização	21
2.3.1	<i>Coefficientes não negativos</i>	22
2.3.2	<i>Extensão para coeficientes arbitrários</i>	25
2.4	Suavização Hiperbólica	37
3	CORTE MÁXIMO (MAX-CUT)	41
3.1	Formulações Matemáticas	42
3.1.1	<i>Formulação por restrições triangulares</i>	42
3.1.2	<i>Formulação binária $\{0, 1\}$</i>	44
3.1.3	<i>Formulação binária $\{-1, 1\}$</i>	45
3.2	Indexação nos Vértices	45
3.2.1	<i>Condições de otimalidade</i>	46
3.2.2	<i>Fixação de variáveis</i>	48
3.2.3	<i>Desigualdades válidas</i>	50
3.2.4	<i>Cortes lógicos</i>	51
3.3	Indexação nas Arestas	53
3.3.1	<i>Condições de otimalidade</i>	53
3.3.2	<i>Desigualdades válidas</i>	54
3.4	Aplicação das Linearizações Clássica e de Glover	55
3.5	Aplicação da t -Linearização ao Max-Cut	57
3.6	Aplicação da Suavização Hiperbólica ao Max-Cut	58
3.7	Heurísticas para o Max-Cut	60
3.7.1	<i>Breve revisão bibliográfica</i>	60
3.7.2	<i>Heurística gulosa com busca local</i>	62
3.7.3	<i>Heurística baseada em condições de otimalidade</i>	65
3.8	Experimentos Computacionais e Comentários	66
3.8.1	<i>Comparação entre os limites superiores</i>	67
3.8.2	<i>Comparação entre os limites inferiores</i>	68
3.8.3	<i>Força das desigualdades</i>	70
3.9	Conclusão	72
4	PROBLEMA DA DIVERSIDADE MÁXIMA	75
4.1	Formulações Lineares	76
4.2	Aplicação da t -Linearização ao PDM	77

4.3	Fortalecimento das Restrições t -Linearizadas	78
4.4	Propriedades das Soluções Ótimas	79
4.4.1	<i>Fixação de variáveis</i>	79
4.4.2	<i>Restrições válidas</i>	82
4.5	Heurística com Ordenação e Busca Local	84
4.6	Experimentos Computacionais	84
4.6.1	<i>Limite Superior</i>	85
4.6.2	<i>Um algoritmo branch and bound</i>	88
4.7	Conclusão	93
5	CLIQUE MÁXIMA PONDERADA EM ARESTA	95
5.1	Formulações	96
5.2	Aplicação da t -Linearização ao PCMPA	97
5.3	Fortalecimento das Restrições t -Linearizadas	99
5.4	Propriedades das Soluções Ótimas	101
5.4.1	<i>Fixação de variáveis</i>	101
5.4.2	<i>Restrições válidas</i>	103
5.5	Heurística Gulosa	104
5.6	Experimentos Computacionais	105
5.6.1	<i>Limite superior</i>	106
5.6.2	<i>Um algoritmo branch and bound</i>	108
5.7	Conclusão	113
6	CONCLUSÃO	115
6.1	Trabalhos Futuros	116
	REFERÊNCIAS	117

1 INTRODUÇÃO

A área de Otimização Combinatória engloba uma grande diversidade de problemas práticos que buscam por soluções que façam melhor uso dos recursos envolvidos. Nesses problemas, o interesse é encontrar a solução que otimize os objetivos de interesse e respeite as restrições impostas. Em geral, os objetivos consistem em aproveitar melhor os materiais no processo de produção, otimizar o tempo para realizar certas ações e operações, transportar materiais pelas melhores rotas, aumentar lucros e diminuir gastos etc (Miyazawa e de Souza, 2015). Para ilustrar a diversidade da área, vários problemas encontram aplicações em escalonamento de tarefas (Giffler e Thompson, 1960), localização de centros distribuidores (Church e Reville, 1974), roteamento de veículos (Toth e Vigo, 2014), projeto de circuitos VLSI (Barahona *et al.*, 1988), sequenciamento de DNA (Kececioglu e Myers, 1995) etc.

Os problemas de otimização combinatória podem ser de minimização ou de maximização de uma função, aplicada a um conjunto finito, que em geral é enumerável, onde se deseja obter a melhor solução possível. Apesar de finito, métodos ingênuos de enumerar todas as possíveis soluções são praticáveis apenas para tamanho de instâncias consideradas pequenas (Wolsey, 1998). Portanto, para instâncias de tamanhos moderados e grandes, faz-se necessário o uso métodos mais elaborados para encontrar a melhor solução possível.

Uma das formas de se entender e resolver problemas de otimização combinatória é através do uso de modelos matemáticos, tais como de Programação Semidefinida (PSD), Programação Linear (PL), Programação Inteira (PI), Programação Inteira Mista (Linear) (PIM), Programação Binária etc. Sendo o modelo mais geral, a programação semidefinida consiste em minimizar ou maximizar uma função linear sujeito à restrição de que matrizes envolvidas na formulação do problema sejam simétricas e semidefinidas positivas (Vandenberghe e Boyd, 1996). O modelo mais particular, a programação linear, consiste em maximizar ou minimizar uma função linear definida sobre um conjunto de variáveis, as quais devem satisfazer um sistema de equações e inequações lineares. Já a programação inteira, inteira mista e inteira binária, consistem basicamente da programação linear com acréscimo, respectivamente, da restrição de que algumas variáveis sejam inteiras, todas sejam inteiras e todas inteiras do tipo 0 – 1 (Ferreira *et al.*, 2001).

Problemas de otimização combinatória podem ser igualmente formulados como problemas de programação inteira binária (Wolsey, 1998). Há, porém, vários problemas desse área onde a modelagem mais natural leva a uma formulação de programação não linear com variáveis binárias. Este é o caso, por exemplo, de problemas conhecidos, tais como: atribuição (Loiola *et al.*, 2007; O'Kelly, 1987), agendamento (Ullman, 1975; Warner e Prawda, 1972), diversidade máxima (Martí, Gallego, e Duarte, 2010; Kuo, Glover, e Dhir, 1993), corte máximo (Goemans e Williamson, 1995; Rendl, Rinaldi, e Wiegele,

2010) etc.

A não linearidade em problemas de programação inteira é normalmente tratada com o uso de técnicas envolvendo uma aproximação linear por partes (Dantzig, 1963; Hu, 1969) ou a transformação da função não linear em uma função polinomial, a seguir convertida em uma função linear de variáveis 0 – 1 (Balas, 1964; Hammer e Rudeanu, 1968). Na maioria das vezes, a não linearidade em problemas de programação inteira aparece já na forma polinomial, sendo que um número significativo dos casos envolve apenas termos de segunda ordem (Glover, 1975).

Na literatura de programação matemática, costuma-se chamar de programação quadrática a subárea que engloba problemas de otimização consistindo na maximização/minimização de uma função quadrática sujeita a restrições expressas por igualdades/desigualdades lineares. Particularmente, quando as variáveis do problema estão restritas a valores 0 – 1, encontramos no escopo da programação quadrática binária. Essa área engloba muitos dos problemas NP-Difíceis mais desafiadores em otimização combinatória (Gary e Johnson, 1979).

Devido à ampla gama de problemas quadráticos binários da literatura e à dificuldade em resolvê-los de forma exata, faz-se necessário a obtenção de limites (inferior e superior) que possam auxiliar o desenvolvimento de métodos exatos. Gueye e Michelon (2009) dividiram esses limites em 04 grupos, de acordo com as técnicas usadas para obtê-los: programação semidefinida (Goemans e Williamson, 1995; Rendl, Rinaldi, e Wiegele, 2010), decomposição lagrangeana (Chardaire e Sutter, 1995; Elloumi, Faye, e Soutif, 2000), métodos posiform (Boros e Hammer, 2002; Billionnet e Sutter, 1994) e técnicas de linearização (Fortet, 1959b,a; Rodrigues, 2010). Acrescentamos a esse grupo a suavização hiperbólica, técnica amplamente aplicada a problemas contínuos não diferenciáveis (Xavier, 2010), (Xavier e Xavier, 2011), (Xavier *et al.*, 2014), (Souza *et al.*, 2011), (Bagirov *et al.*, 2015), (Xavier e Xavier, 2016), mas com potencial de aplicação também a problemas quadráticos binários.

Dentre as técnicas de linearização para problemas quadráticos binários, que consistem em transformar a função quadrática em uma função linear, Rodrigues *et al.* (2012) destacam duas abordagens: (i) linearização com acréscimo de um número polinomial de variáveis e restrições lineares; (ii) linearização sem adição de variáveis, mas com um número exponencial de restrições lineares. No primeiro grupo, encontra-se o procedimento clássico de linearização (Fortet, 1959b) e a linearização proposta por Glover (1975); no segundo grupo, está a t -linearização, proposta no contexto do Problema Quadrático da Mochila (PQM) (Rodrigues, 2010; Rodrigues *et al.*, 2012).

A t -linearização é uma abordagem recente e ainda pouco estudada, que consiste basicamente de duas etapas. Primeiro, substitui-se o termo quadrático da função objetivo por uma variável real t , que é limitada superiormente pela expressão quadrática, com a inclusão de uma restrição adicional. Depois, essa restrição quadrática é substituída por

um conjunto exponencial de restrições lineares, equivalentes quando os coeficientes do termo quadrático são não negativos (Rodrigues *et al.*, 2012).

A suavização hiperbólica é uma técnica de penalização proposta para resolver problemas não diferenciáveis que tem obtido bons resultados computacionais (Xavier, 2010; Xavier e Xavier, 2016; Bagirov, Al Nuaimat, e Sultanova, 2013; Xavier e Xavier, 2011). Ela consiste em substituir a resolução direta do problema não diferenciável pela resolução de uma sequência de problemas diferenciáveis, em princípio mais fáceis, e que gradativamente se aproximam do problema original (Lavor, 2004). Cada subproblema, por ser infinitamente diferenciável, torna viável sua resolução através de métodos de otimização mais robustos, que são fundamentados em informações do gradiente, tais como método do gradiente conjugado, quase-Newton ou Newton (Xavier e Xavier, 2016).

Neste trabalho, concentramo-nos em utilizar técnicas ainda pouco exploradas para tratar problemas quadráticos 0 – 1, em particular t -linearização e suavização hiperbólica. O problema quadrático (PQ), base para o estudo de outros problemas dessa classe, pode ser escrito da seguinte forma:

$$(PQ) \quad \max \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} x_i x_j + \sum_{i=1}^n b_i x_i$$

$$s.a \quad x \in \{0, 1\}^n,$$

onde c_{ij} , $1 \leq i < j \leq n$, são coeficientes reais arbitrários. Por conveniência, definimos os coeficientes adicionais $c_{ij} = c_{ji}$ e $c_{ii} = 0$, $1 \leq j \leq i \leq n$. A esse problema base consideramos o acréscimo de restrições lineares, em particular restrições de cardinalidade.

O restante desse trabalho contém 5 capítulos e está organizado da seguinte forma. No Capítulo 2 iremos abordar o conceito de linearização (Clássica, Glover e t -linearização) em problemas quadráticos binários. Vamos apresentar provas alternativas para o desenvolvimento realizado por Rodrigues (2010) para (PQ) com coeficientes não negativos, que serão base para a extensão do método da t -linearização para (PQ) quando os coeficientes da função objetivo são arbitrários. Dessa maneira, generalizamos os resultados obtidos em (Rodrigues, 2010) e (Rodrigues *et al.*, 2012). Vamos mostrar que a t -linearização oferece melhor limite superior dentre as linearizações aqui estudadas. Ainda no Capítulo 2 vamos revisar o arcabouço da suavização hiperbólica, usado, por exemplo, em (Xavier e Xavier, 2016).

Nos capítulos seguintes, usamos essas duas metodologias gerais acopladas a outros ingredientes para resolver problemas quadráticos binários clássicos, procurando explorar propriedades específicas dos mesmos. No Capítulo 3, consideramos o clássico problema de corte máximo (max-cut) em um grafo não direcionado. Vamos mostrar que existem diferentes formulações matemáticas para o problema e como podem ser convertidas uma na outra. Ainda nesse capítulo, vamos mostrar pela primeira vez a aplicação

da t -linearização quando consideramos a formulação quadrática do problema do corte máximo. Experimentos computacionais mostram que a t -linearização requer menos restrições para alcançar o mesmo limite obtido pelas linearizações clássica e Glover. Ainda nesse capítulo, vamos mostrar como obter uma versão suavizada do modelo irrestrito definido sobre variáveis indexadas pelos vértices, porém fortalecido pelas restrições triangulares, que são expressas independentemente em função das variáveis indexadas pelas arestas. Experimentos atestam melhores limites quando a suavização é fortalecida pelas restrições triangulares e incorpora um algoritmo de busca local. Derivamos ainda condições de otimalidade que permitem a expressão de novas desigualdades válidas para o problema, o estabelecimento de teoremas para fixação de variáveis, além da proposição de heurísticas gulosas simples. Todos esses ingredientes que propomos são combinados de diferentes formas na construção de um método de solução exata para o max-cut. Por fim, fechamos o capítulo com comentários e conclusões.

O Capítulo 4 é dedicado ao problema da diversidade máxima (PDM). Vamos revisar suas principais formulações e, em seguida, mostrar a aplicação da t -linearização ao problema. Inspirados em Rodrigues (2010), demonstramos um teorema de *lifting* para fortalecer as restrições da t -linearização, usando a restrição de cardinalidade do problema (PDM). Experimentos computacionais com classes de instâncias da literatura atestam que as restrições t -linearizadas fortalecidas obtêm os melhores limites dentre as linearizações comparadas. Apresentamos também o desenvolvimento de novas restrições válidas, a partir de generalizações de propriedades de poda inicialmente mostradas em Martí, Gallego, e Duarte (2010). Ainda nesse capítulo, empregamos tal desenvolvimento teórico para proposição de um método que resolve o problema de forma exata e realizamos outros experimentos computacionais, comparando os resultados com aqueles do melhor método exato disponível. Fechamos o capítulo com comentários e conclusões.

No Capítulo 5, estudamos o problema da clique máxima ponderada em arestas (PCMPA). Vamos revisar as principais formulações do problema e, em seguida, mostrar a aplicação da t -linearização ao mesmo. Generalizamos o teorema de *lifting* apresentado para o (PDM), de modo a fortalecer as restrições da t -linearização quando consideramos custos arbitrários. Experimentos computacionais com classes de instâncias da literatura atestam que as restrições t -linearizadas fortalecidas obtêm os melhores limites, em instâncias com custos arbitrários, dentre as linearizações comparadas. Apresentamos também a generalização do teorema de novas restrições válidas, que foi mostrado para o PDM. Ainda nesse capítulo, empregamos tal desenvolvimento teórico para proposição de um método que resolve o problema de forma exata e realizamos experimentos computacionais com classes de instâncias da literatura, comparando os resultados com as principais formulações do problema. Fechamos o capítulo com comentários e conclusões. No último Capítulo, concluímos esta tese mostrando o estado atual da nossa pesquisa e quais os principais direcionamentos que pretendemos seguir.

2 METODOLOGIA

Neste capítulo, vamos abordar o problema quadrático binário

$$(PQ) \quad \max \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} x_i x_j + \sum_{i=1}^n b_i x_i = \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji} x_i x_j + \sum_{i=1}^n b_i x_i \quad (1)$$

s.a $x \in \{0, 1\}^n$,

onde c_{ij} , $1 \leq i < j \leq n$, são coeficientes reais arbitrários. Por conveniência, definimos os coeficientes adicionais $c_{ij} = c_{ji}$ e $c_{ii} = 0$, $1 \leq j \leq i \leq n$.

Nas três próximas subseções, vamos apresentar três maneiras de obter uma formulação linear equivalente, duas delas bastante conhecidas na literatura. A primeira, a que nos referiremos como linearização clássica, obtém um modelo linear 0 – 1, com um número quadrático (em n) de variáveis e restrições; a segunda, proposta por Glover (1975), leva a um problema linear 0 – 1 misto, definido a partir das mesmas variáveis binárias de (1), junto com um número linear de variáveis contínuas e restrições. A terceira, chamada t -linearização, foi proposta mais recentemente e usa, além das variáveis binárias originais, apenas uma variável contínua; porém, demanda um número exponencial de restrições, que podem ser separadas polinomialmente. Ela foi mostrada correta quando os coeficientes c_{ij} são não negativos. Aqui mostramos como aplicá-la com coeficientes arbitrários, generalizando resultados da literatura, que incluem tanto a equivalência com o modelo original (1), como também a existência de algoritmo polinomial para separação das restrições. Parte desses resultados compõe um artigo publicado nos anais no Simpósio Brasileiro de Pesquisa Operacional - SBPO 2017 (Soares *et al.*, 2017).

Na última seção, descrevemos brevemente o método de suavização hiperbólica, inicialmente proposto para solução de problemas contínuos não diferenciáveis. Em nosso trabalho, ele será usado para obter boas soluções de problemas quadráticos binários.

2.1 Linearização Clássica

A técnica de linearização para um problema quadrático binário consiste em transformar a função quadrática em uma função linear, podendo ser dividida em duas abordagens, sendo elas: (i) linearização sem adição de variáveis, mas com um número exponencial de restrições; (ii) linearização com um número polinomial de variáveis e restrições (Rodrigues *et al.*, 2012). No contexto dessa segunda abordagem, em Fortet (1959b) e Fortet (1959a), o autor mostra como converter um problema de programação polinomial 0 – 1 em um problema de programação linear 0 – 1. Esse procedimento, conhecido como linearização clássica, consiste em substituir cada produto $x_i x_j$ por uma variável não negativa y_{ij} e adicionar restrições $y_{ij} \leq x_i$, $y_{ij} \leq x_j$, $y_{ij} \geq x_i + x_j - 1$.

Dessa forma, podemos aplicar a técnica de linearização clássica ao problema

(1) e reescrevê-lo como:

$$\begin{aligned}
(PQ)_{lc} \quad & \max \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} y_{ij} + \sum_{i=1}^n b_i x_i \\
s.a \quad & y_{ij} \leq x_i, \quad \forall 1 \leq i < j \leq n, \\
& y_{ij} \leq x_j, \quad \forall 1 \leq i < j \leq n, \\
& y_{ij} \geq x_i + x_j - 1 \quad \forall 1 \leq i < j \leq n, \\
& y_{ij} \in \{0, 1\} \quad \forall 1 \leq i < j \leq n, \\
& x \in \{0, 1\}^n.
\end{aligned} \tag{2}$$

Note que podemos substituir $y_{ij} \in \{0, 1\}$ por $0 \leq y_{ij} \leq 1$ em (2).

Para um único termo quadrático, a linearização clássica é apertada, como mostra o seguinte resultado.

Proposição 2.1. *Se $A = \text{conv}\{(x_1, x_2, y) \in \mathbb{B}^3 : y = x_1 x_2\}$ e $B = \{(x_1, x_2, y) \in \mathbb{R}^3 : y \leq x_1, y \leq x_2, y \geq x_1 + x_2 - 1, y \geq 0, x_1 \leq 1, x_2 \leq 1\}$, então $A = B$.*

Demonstração. Claramente $A \subseteq B$. Considere agora $(\bar{x}_1, \bar{x}_2, \bar{y}) \in B$. Temos então $(\bar{x}_1, \bar{x}_2, \bar{y}) = (\bar{x}_1 - \bar{y})(1, 0, 0) + (\bar{x}_2 - \bar{y})(0, 1, 0) + \bar{y}(1, 1, 1) + (1 + \bar{y} - \bar{x}_1 - \bar{x}_2)(0, 0, 0)$, mostrando que $(\bar{x}_1, \bar{x}_2, \bar{y}) \in A$. \square

Entretanto, é sabido que a relaxação linear de (2) torna-se fraca quando há muitos termos quadráticos.

2.2 Linearização de Glover

Glover e Woolsey (1974) propõem uma outra linearização para (PQ). A ideia é definir variáveis contínuas $u \in \mathbb{R}^{n-1}$ tais que

$$u_i = x_i \sum_{j=i+1}^n c_{ij} x_j \quad \forall i \in \{1, \dots, n-1\}. \tag{3}$$

Para isso, seja

$$\bar{U}_i = \sum_{j=i+1}^n c_{ij}^+ \quad \text{e} \quad \underline{U}_i = \sum_{j=i+1}^n c_{ij}^-,$$

onde $c_{ij}^+ = \max(0, c_{ij})$ e $c_{ij}^- = \min(0, c_{ij})$. Note que $\underline{U}_i \leq \sum_{j=i+1}^n c_{ij} x_j \leq \bar{U}_i$ e $\underline{U}_i \leq u_i \leq \bar{U}_i$.

Podemos então reescrever (PQ) como

$$(PQ)_{lg} \quad \max \quad \sum_{i=1}^{n-1} u_i + \sum_{i=1}^n b_i x_i \quad (4)$$

$$s.a \quad u_i \leq \bar{U}_i x_i, \quad 1 \leq i \leq n-1, \quad (5)$$

$$u_i \leq \sum_{j=i+1}^n c_{ij} x_j - \underline{U}_i (1 - x_i), \quad 1 \leq i \leq n-1, \quad (6)$$

$$x \in \{0, 1\}^n. \quad (7)$$

Quando $x_i = 1$, a restrição (5) torna-se redundante enquanto (6) leva a $u_i \leq \sum_{j=i+1}^n c_{ij} x_j$. No ótimo, temos a igualdade, o que equivale a (3) para $x_i = 1$. Quando $x_i = 0$, (5) estabelece que $u_i \leq 0$, levando (6) a se tornar redundante. Além disso, a função (4) garante $u_i = 0$ no ótimo. Portanto (5)–(7) garantem a igualdade (3) no ótimo.

O modelo linearizado (4)–(7) possui menos variáveis que (2), entretanto sua relaxação pode ser mais fraca.

Teorema 2.1. *O valor da relaxação de (4)–(7) é pelo menos aquele da relaxação de (2).*

Demonstração. Seja (x, y) viável para a relaxação de (2). Defina $u_i = \sum_{j=i+1}^n c_{ij} y_{ij}$. O valor da função objetivo (4) em (x, u) é trivialmente igual ao valor de objetivo de (2) em (x, y) . Então, é suficiente mostrar que (x, u) satisfaz (5)–(6). Por um lado, temos que

$$u_i = \sum_{j=i+1}^n c_{ij} y_{ij} \leq \sum_{j=i+1}^n c_{ij}^+ y_{ij} \leq \sum_{j=i+1}^n c_{ij}^+ x_j.$$

Por outro lado,

$$\begin{aligned} u_i &= \sum_{j=i+1}^n c_{ij} y_{ij} = \sum_{j=i+1}^n c_{ij}^+ y_{ij} + \sum_{j=i+1}^n c_{ij}^- y_{ij} \\ &\leq \sum_{j=i+1}^n c_{ij}^+ x_j + \sum_{j=i+1}^n c_{ij}^- (x_i + x_j - 1) = \sum_{j=i+1}^n c_{ij} x_j - \sum_{j=i+1}^n c_{ij}^- (1 - x_i). \end{aligned}$$

□

2.3 t -Linearização

Em Rodrigues (2010) e Rodrigues *et al.* (2012), os autores propuseram um *framework* para linearizar o Problema Quadrático da Mochila (PQM), com o acréscimo de uma única variável, mas com adição de um número exponencial de restrições. Essa abordagem consiste basicamente de duas etapas, descritas a seguir. Na primeira etapa, os autores trocam o termo quadrático da função objetivo por uma variável real t e adicionam uma restrição limitando superiormente o valor dessa variável ao valor da expressão quadrática. A segunda etapa consiste em substituir a restrição quadrática por um conjunto de restrições

lineares, derivadas da caracterização do fecho convexo do conjunto definido pela restrição quadrática e restrições de integralidade (Rodrigues, 2010). No caso do (PQM), os coeficientes do termo quadrático são não negativos. Essa propriedade é fortemente usada no desenvolvimento de Rodrigues *et al.* (2012).

Vamos estender a teoria da t -linearização e aplicá-la ao problema (1) com coeficientes arbitrários associados aos termos quadráticos. Para facilitar o entendimento, primeiro apresentamos, na Subseção (2.3.1), a t -linearização para o caso onde os coeficientes do termo quadrático são não negativos, seguindo Rodrigues *et al.* (2012). Na subseção seguinte, estendemos o desenvolvimento para custos arbitrários.

2.3.1 Coeficientes não negativos

Ao longo desta subseção, consideramos o problema (PQ) e admitimos que $c_{ij} = c_{ji} \geq 0 \forall 1 \leq i \leq j \leq n$. De acordo com a primeira etapa de Rodrigues (2010), reescrevemos (1) com o auxílio de uma variável t , gerando

$$(PQ)_t \quad \max \quad t + \sum_{i=1}^n b_i x_i$$

$$s.a \quad t \leq \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji} x_i x_j,$$

$$(x, t) \in \mathbb{B}^n \times \mathbb{R}.$$

Para descrever a segunda etapa, sejam

$$Z = \left\{ (x, t) \in \mathbb{B}^n \times \mathbb{R} : t \leq \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji} x_i x_j \right\}$$

o conjunto viável de $(PQ)_t$ e S_n o conjunto de todas as permutações de $\{1, \dots, n\}$. O conjunto Z pode ser expresso por restrições lineares, como segue:

Teorema 2.2. *Se $c_{ij} \geq 0$, para todo $1 \leq i, j \leq n$, então $Z = Z'$, onde*

$$Z' = \left\{ (x, t) \in \mathbb{B}^n \times \mathbb{R} : t \leq \sum_{i=1}^n \sum_{j=1}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(i)} x_{\pi(j)} \quad \forall \pi \in S_n \right\}.$$

Demonstração. Primeiro, note que

$$\sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji} x_i x_j = \sum_{i=1}^n \sum_{j=i}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(i)} x_{\pi(j)} \quad \forall \pi \in S_n. \quad (8)$$

Se $(x, t) \in Z$, então temos que $(x, t) \in Z'$ devido a (8) e

$$\sum_{i=1}^n \sum_{j=1}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(i)} x_{\pi(j)} \leq \sum_{i=1}^n \sum_{j=1}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(i)}.$$

Suponha agora $(x, t) \in Z'$ e seja $\pi \in S_n$ a permutação que ordena as componentes de x em ordem não crescente, ou seja, $x_{\pi(j)} \geq x_{\pi(i)}$ se $j \leq i$. Observe que $(x_{\pi(1)}, \dots, x_{\pi(n)}) = (1, \dots, 1, 0, \dots, 0)$. Como $(x, t) \in Z'$, temos que

$$t \leq \sum_{i=1}^n \sum_{j=1}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(i)} = \sum_{i=1}^n \sum_{j=1}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(i)} x_{\pi(j)},$$

onde a igualdade decorre da ordenação das entradas de x por π . \square

Pelo Teorema (8), podemos substituir Z por Z' na descrição de $(PQ)_t$, obtendo

$$\begin{aligned} (PQ)_t^\pi \quad \max \quad & t + \sum_{i=1}^n b_i x_i \\ \text{s.a} \quad & t \leq \sum_{i=1}^n \sum_{j=1}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(i)} \quad \forall \pi \in S_n, \\ & (x, t) \in \mathbb{B}^n \times \mathbb{R}, \end{aligned} \quad (9)$$

e concluindo assim a segunda etapa.

Note que $(PQ)_t^\pi$ é um modelo linear inteiro com essencialmente as mesmas variáveis que (PQ) , mas com um número exponencial de restrições. Felizmente, a separação das restrições lineares que definem Z' pode ser feita em tempo polinomial, como mostraremos a seguir.

Teorema 2.3. *Seja $\hat{x} \in [0, 1]^n$. Se $c_{ij} \geq 0$, para todo $1 \leq i, j \leq n$, então uma solução do problema*

$$\min_{\pi \in S_n} \sum_i \sum_{j < i} c_{\pi(j)\pi(i)} \hat{x}_{\pi(i)} \quad (10)$$

ocorre na permutação $\hat{\pi} \in S_n$ que ordena as componentes de \hat{x} em ordem não crescente, ou seja, tal que

$$\hat{x}_{\hat{\pi}(i)} \geq \hat{x}_{\hat{\pi}(j)} \quad \forall i, j \quad i \leq j \leq n. \quad (11)$$

Demonstração. Seja $\pi \in S_n$ uma solução ótima para (10). Suponha que π não satisfaz a propriedade (11). Seja então $l < n$ o maior inteiro tal que as primeiras l posições de \hat{x} , segundo π , estão ordenadas, ou seja,

$$\hat{x}_{\pi(1)} \geq \hat{x}_{\pi(2)} \geq \dots \geq \hat{x}_{\pi(l)} < \hat{x}_{\pi(l+1)}.$$

Seja $k \leq l$ o menor inteiro satisfazendo $\hat{x}_{\pi(l+1)} > \hat{x}_{\pi(k)}$, de modo que

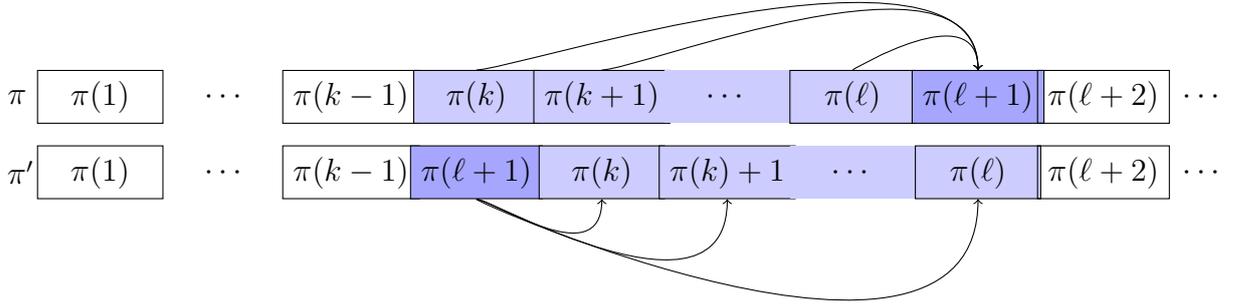
$$\hat{x}_{\pi(1)} \geq \dots \geq \hat{x}_{\pi(k-1)} \geq \hat{x}_{\pi(l+1)} > \hat{x}_{\pi(k)} \geq \hat{x}_{\pi(l)}. \quad (12)$$

Defina uma nova permutação π' tal que

$$\pi'(i) = \begin{cases} \pi(i), & i = 1 \dots k-1 \text{ ou } i = l+2 \dots n, \\ \pi(l+1), & i = k, \\ \pi(i-1), & i = k+1 \dots l+1. \end{cases}$$

Ou seja, π' ordena as entradas de \hat{x} como em (12), vê Figura 1.

Figura 1: Diferença entre $\Theta := \sum_i \sum_{j<i} c_{\pi(j)\pi(i)} \hat{x}_{\pi(i)}$ e $\Theta' := \sum_i \sum_{j<i} c_{\pi'(j)\pi'(i)} \hat{x}_{\pi'(i)}$. Em azul as posições diferentes entre π e π' . As setas indicam os coeficientes que aparecem Θ mas não em Θ' (figura superior) e vice-versa (figura inferior)



Fonte: Elaborado pelo autor.

Então

$$\begin{aligned} \sum_i \sum_{j<i} c_{\pi'(j)\pi'(i)} \hat{x}_{\pi'(i)} - \sum_i \sum_{j<i} c_{\pi(j)\pi(i)} \hat{x}_{\pi(i)} &= \sum_{i=k}^l c_{\pi(l+1)\pi(i)} \hat{x}_{\pi(i)} - \sum_{i=k}^l c_{\pi(i)\pi(l+1)} \hat{x}_{\pi(l+1)} \\ &= \sum_{i=k}^l \underbrace{c_{\pi(i)\pi(l+1)}}_{\geq 0} \underbrace{(\hat{x}_{\pi(i)} - \hat{x}_{\pi(l+1)})}_{< 0} \\ &\leq 0. \end{aligned}$$

Logo, π' também é ótimo para (10) e satisfaz a propriedade (11) até pelo menos $l+1$, ou seja,

$$\hat{x}_{\pi'(1)} \geq \hat{x}_{\pi'(2)} \geq \dots \geq \hat{x}_{\pi'(l)} \geq \hat{x}_{\pi'(l+1)}$$

Repetindo o processo a partir de π' , em um número fixo de passos obtemos uma permutação $\hat{\pi}$ satisfazendo (11) e que também é ótima. \square

Na verdade, Rodrigues (2010) mostra que a integralidade das variáveis x pode ser descartada em $(PQ)_t^\pi$, pois sua relaxação linear fornece o ótimo, conforme teorema abaixo. Para demonstrá-lo, introduzimos a seguinte notação. Dado $\hat{x} \in [0, 1]^n$ tal que $\hat{x}_1 \geq \hat{x}_2 \geq \dots \geq \hat{x}_n$, seja $t_{\hat{x}}$ o valor ótimo do problema (10). Pelo Teorema 2.3, deduzimos que $t_{\hat{x}} = \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji} \hat{x}_i$. Para $k \in \{0, \dots, n\}$, vamos denotar por $e^k \in \mathbb{B}^n$ o vetor binário onde exatamente as primeiras k componentes são iguais a 1.

Teorema 2.4. Se $c_{ij} \in \mathbb{Q}_+$, para todo $1 \leq i, j \leq n$, então $\text{conv}(Z') = \bar{Z}$, onde $\bar{Z} = \left\{ (x, t) \in [0, 1]^n \times \mathbb{R} : t \leq \sum_{i=1}^n \sum_{j=1}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(i)} \forall \pi \in S_n \right\}$.

Demonstração. Trivialmente, $\text{conv}(Z') \subseteq \bar{Z}$. Para mostrar a inclusão inversa, é suficiente mostrar que todo vértice de \bar{Z} está em $\text{conv}(Z')$. Seja (x, t) vértice de \bar{Z} . Sem perda de generalidade, admita que $1 \geq x_1 \geq x_2 \geq \dots \geq x_n \geq 0$. Observe que $t \leq t_x$, $(x, t_x) \in \bar{Z}$ e $(0, -1)$ é direção de recessão de \bar{Z} . Se $t < t_x$, temos que $(x, t) = (x, t_x) + (t_x - t)(0, -1)$, mostrando que (x, t) não é vértice. Logo, $t = t_x$. Temos também que $x = \sum_{k=1}^{n-1} e^k (x_k - x_{k+1}) + e^n x_n$. Então, definindo

$$\alpha_k = \begin{cases} x_k - x_{k+1}, & \text{se } k \in \{1, \dots, n-1\}, \\ x_n, & \text{se } k = n, \\ 1 - x_1, & \text{se } k = 0, \end{cases}$$

temos que

$$x = \sum_{k=0}^n \alpha_k e^k, \quad \sum_{k=0}^n \alpha_k = 1, \quad \alpha_k \geq 0 \forall k \in \{0, \dots, n\}.$$

Note também que $(e^k, t_{e^k}) \in Z' \subset \bar{Z}$, onde $t_{e^k} = \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji} e_i^k$. Logo,

$$t_x = \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji} x_i = \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji} \sum_{k=0}^n \alpha_k e_i^k = \sum_{k=0}^n \alpha_k \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji} e_i^k = \sum_{k=0}^n \alpha_k t_{e^k}.$$

Das duas expressões destacadas acima, concluimos que (x, t_x) é combinação convexa de pontos de Z' . \square

Corolário 2.1. Se $c_{ij} \in \mathbb{Q}_+$, para todo $1 \leq i, j \leq n$, então a formulação (9) e sua relaxação linear possuem o mesmo valor ótimo.

2.3.2 Extensão para coeficientes arbitrários

Seja

$$F(x) = \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji} x_i x_j + \sum_{i=1}^n b_i x_i. \quad (13)$$

Antes de aplicarmos as etapas descritas por Rodrigues (2010), vamos dividir os coeficientes de acordo com o seu sinal. Denote $d^+ = \max\{0, d\}$, $d^- = \min\{0, d\}$, $\forall d \in \mathbb{R}$, e $\bar{a} = 1 - a$, $\forall a \in [0, 1]$. Observe que $d = d^+ + d^-$.

Podemos reescrever $F(x)$ como

$$\begin{aligned}
F(x) &= \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji}^+ x_i x_j + \sum_{i=1}^n \sum_{j=1}^{i-1} \underbrace{\frac{1}{2} c_{ji}^- x_i (1 - \bar{x}_j) + \frac{1}{2} c_{ji}^- (1 - \bar{x}_i) x_j}_{c_{ji}^- x_i x_j} + \sum_{i=1}^n b_i x_i \\
&= \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji}^+ x_i x_j + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^{i-1} -c_{ji}^- (x_i \bar{x}_j + \bar{x}_i x_j) + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji}^- (x_i + x_j) + \sum_{i=1}^n b_i x_i \\
&= Q(x) + L(x) + \sum_{i=1}^n b_i x_i,
\end{aligned}$$

onde

$$\begin{aligned}
Q(x) &= \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji}^+ x_i x_j + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^{i-1} -c_{ji}^- (x_i \bar{x}_j + \bar{x}_i x_j), \\
L(x) &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji}^- (x_i + x_j) = \frac{1}{2} \sum_i \sum_{j < i} c_{ji}^- x_i + \frac{1}{2} \sum_i \sum_{j > i} c_{ji}^- x_i = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n c_{ij}^- x_i.
\end{aligned}$$

Agora, aplicando a etapa 1, podemos reescrever (PQ) como

$$\begin{aligned}
(PQ)_{\pm t} \quad \max \quad & t + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n c_{ij}^- x_i + \sum_{i=1}^n b_i x_i \\
\text{s.a} \quad & t \leq \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji}^+ x_i x_j + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^{i-1} -c_{ji}^- (x_i \bar{x}_j + \bar{x}_i x_j), \\
& (x, t) \in \mathbb{B}^n \times \mathbb{R}, \quad \bar{x}_j = 1 - x_j \quad \forall j = 1, \dots, n.
\end{aligned}$$

Os problemas (PQ) e $(PQ)_{\pm t}$ são equivalentes. A função objetivo em $(PQ)_{\pm t}$ possui agora uma expressão linear, ao passo que uma nova restrição quadrática foi adicionada ao problema. A seguir vamos mostrar como substituí-la por um conjunto de restrições lineares, funcionalmente equivalentes. Seja

$$Z = \left\{ (x, t) \in \mathbb{B}^n \times \mathbb{R} : t \leq \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji}^+ x_i x_j + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^{i-1} -c_{ji}^- (x_i \bar{x}_j + \bar{x}_i x_j) \right\}$$

o conjunto viável de $(PQ)_{\pm t}$. Denote por $\mathcal{N} = \{(N_0, N_1) : N_0 \cup N_1 = N, N_0 \cap N_1 = \emptyset\}$ o conjunto de partições de $N = \{1, \dots, n\}$ e S_n o conjunto de todas as permutações de $\{1, \dots, n\}$. O conjunto Z pode ser expresso por restrições lineares, como segue:

Teorema 2.5. *O conjunto Z é igual a*

$$Z' := \left\{ (x, t) \in \mathbb{B}^n \times \mathbb{R} : t \leq \sum_{i \in N} \sum_{\substack{\pi(j) \in N_1 \\ j < i}} c_{\pi(j)\pi(i)}^+ x_{\pi(i)} + \sum_{i \in N} \sum_{\substack{\pi(j) \in N_0 \\ j < i}} c_{\pi(j)\pi(i)}^+ x_{\pi(j)} + \right.$$

$$\left. \frac{1}{2} \sum_{i \in N} \sum_{\substack{\pi(j) \in N_1 \\ j < i}} -c_{\pi(j)\pi(i)}^- (\bar{x}_{\pi(i)} + \bar{x}_{\pi(j)}) + \frac{1}{2} \sum_{i \in N} \sum_{\substack{\pi(j) \in N_0 \\ j < i}} -c_{\pi(j)\pi(i)}^- (x_{\pi(i)} + x_{\pi(j)}) \forall \pi \in S_n \forall (N_0, N_1) \in \mathcal{N} \right\}.$$

Demonstração. Seja $(x, t) \in Z$ e defina $\bar{x} = \mathbf{1} - x$. Então

$$t \leq Q(x) = \underbrace{\sum_{i \in N} \sum_{j < i} c_{ji}^+ x_i x_j}_A + \frac{1}{2} \underbrace{\sum_{i \in N} \sum_{j < i} (-c_{ji}^-) [x_i \bar{x}_j + \bar{x}_i x_j]}_B.$$

Observe que A acumula a soma $c_{ji}^+ x_i x_j$ para todos os pares não ordenados ij com $c_{ij} > 0$, e B acumula a soma $-\frac{1}{2} c_{ji}^- (x_i \bar{x}_j + \bar{x}_i x_j)$ para todos os pares não ordenados ij com $c_{ij} < 0$. Então, para $\pi \in S_n$, temos

$$Q(x) = \underbrace{\sum_{i \in N} \sum_{j < i} c_{\pi(j)\pi(i)}^+ x_{\pi(i)} x_{\pi(j)}}_C + \frac{1}{2} \underbrace{\sum_{i \in N} \sum_{j < i} (-c_{\pi(j)\pi(i)}^-) [x_{\pi(i)} \bar{x}_{\pi(j)} + \bar{x}_{\pi(i)} x_{\pi(j)}]}_D.$$

Para $(N_0, N_1) \in \mathcal{N}$ partição de N , temos

$$\begin{aligned} C &= \sum_{i \in N} \sum_{\substack{j < i \\ \pi(j) \in N_1}} \underbrace{(c_{\pi(j)\pi(i)}^+ x_{\pi(i)})}_{\geq 0} \underbrace{x_{\pi(j)}}_{\leq 1} + \sum_{i \in N} \sum_{\substack{j < i \\ \pi(j) \in N_0}} \underbrace{(c_{\pi(j)\pi(i)}^+ x_{\pi(j)})}_{\geq 0} \underbrace{x_{\pi(i)}}_{\leq 1} \\ &\leq \sum_{i \in N} \sum_{\substack{j < i \\ \pi(j) \in N_1}} c_{\pi(j)\pi(i)}^+ x_{\pi(i)} + \sum_{i \in N} \sum_{\substack{j < i \\ \pi(j) \in N_0}} c_{\pi(j)\pi(i)}^+ x_{\pi(j)} \end{aligned}$$

e

$$\begin{aligned} D &= \frac{1}{2} \sum_{i \in N} \sum_{\substack{j < i \\ \pi(j) \in N_1}} \underbrace{-c_{\pi(j)\pi(i)}^-}_{\geq 0} \underbrace{[x_{\pi(i)} \bar{x}_{\pi(j)} + \bar{x}_{\pi(i)} x_{\pi(j)}]}_{\leq \bar{x}_{\pi(i)} + \bar{x}_{\pi(j)}} + \frac{1}{2} \sum_{i \in N} \sum_{\substack{j < i \\ \pi(j) \in N_0}} \underbrace{-c_{\pi(j)\pi(i)}^-}_{\geq 0} \underbrace{[x_{\pi(i)} \bar{x}_{\pi(j)} + \bar{x}_{\pi(i)} x_{\pi(j)}]}_{\leq x_{\pi(i)} + x_{\pi(j)}} \\ &\leq \frac{1}{2} \sum_{i \in N} \sum_{\substack{j < i \\ \pi(j) \in N_1}} -c_{\pi(j)\pi(i)}^- (\bar{x}_{\pi(i)} + \bar{x}_{\pi(j)}) + \frac{1}{2} \sum_{i \in N} \sum_{\substack{j < i \\ \pi(j) \in N_0}} -c_{\pi(j)\pi(i)}^- (x_{\pi(i)} + x_{\pi(j)}). \end{aligned}$$

Como $t \leq Q(x) = C + D$ e $\pi \in S_n$ e $(N_0, N_1) \in \mathcal{N}$ são arbitrários, concluímos que $(x, t) \in Z'$.

Suponha agora $(x, t) \in Z'$. Seja $\pi \in S_n$ e defina $N_0 = \{j \in N : x_j = 0\}$ e $N_1 = \{j \in N : x_j = 1\}$. Temos que

$$\begin{aligned} Q(x) &= \sum_{i \in N} \sum_{j < i} c_{\pi(j)\pi(i)}^+ x_{\pi(i)} x_{\pi(j)} + \frac{1}{2} \sum_{i \in N} \sum_{j < i} (-c_{\pi(j)\pi(i)}^-) [x_{\pi(i)} \bar{x}_{\pi(j)} + \bar{x}_{\pi(i)} x_{\pi(j)}] \\ &= \sum_{i \in N} \sum_{\substack{j < i \\ \pi(j) \in N_1}} c_{\pi(j)\pi(i)}^+ x_{\pi(i)} + \frac{1}{2} \sum_{i \in N} \sum_{\substack{j < i \\ \pi(j) \in N_1}} -c_{\pi(j)\pi(i)}^- \bar{x}_{\pi(i)} + \frac{1}{2} \sum_{i \in N} \sum_{\substack{j < i \\ \pi(j) \in N_0}} -c_{\pi(j)\pi(i)}^- x_{\pi(i)}. \end{aligned}$$

Além disso, como $(x, t) \in Z'$, então

$$t \leq \frac{1}{2} \sum_{i \in N} \sum_{\substack{\pi(j) \in N_1 \\ j < i}} -c_{\pi(j)\pi(i)}^- \bar{x}_{\pi(i)} + \frac{1}{2} \sum_{i \in N} \sum_{\substack{\pi(j) \in N_0 \\ j < i}} -c_{\pi(j)\pi(i)}^- x_{\pi(i)} + \sum_{i \in N} \sum_{\substack{\pi(j) \in N_1 \\ j < i}} c_{\pi(j)\pi(i)}^+ x_{\pi(i)} = Q(x).$$

Logo, $(x, t) \in Z$. \square

O Teorema 2.5 generaliza, para coeficientes arbitrários, o resultado de Rodrigues *et al.* (2012). Como consequência, podemos transformar o problema quadrático $(PQ)_{\pm t}$ e, portanto (PQ) , em um problema de programação linear inteira equivalente, com a adição de uma variável extra e um número exponencial de $2^n \cdot n!$ restrições.

$$\begin{aligned} (PQ)_{\pm t}^{\pi \mathcal{N}} \max \quad & t + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n c_{ij}^- x_i + \sum_{i=1}^n b_i x_i \\ \text{s.a.} \quad & t \leq \sum_{i \in N} \sum_{\substack{\pi(j) \in N_1 \\ j < i}} c_{\pi(j)\pi(i)}^+ x_{\pi(i)} + \sum_{i \in N} \sum_{\substack{\pi(j) \in N_0 \\ j < i}} c_{\pi(j)\pi(i)}^+ x_{\pi(j)} + \\ & \frac{1}{2} \sum_{i \in N} \sum_{\substack{\pi(j) \in N_1 \\ j < i}} -c_{\pi(j)\pi(i)}^- (\bar{x}_{\pi(i)} + \bar{x}_{\pi(j)}) + \frac{1}{2} \sum_{i \in N} \sum_{\substack{\pi(j) \in N_0 \\ j < i}} -c_{\pi(j)\pi(i)}^- (x_{\pi(i)} + x_{\pi(j)}) \\ & \forall \pi \in S_n \quad \forall (N_0, N_1) \in \mathcal{N}, \\ & (x, t) \in \mathbb{B}^n \times \mathbb{R}, \quad \bar{x}_j = 1 - x_j \quad \forall j = 1, \dots, n. \end{aligned} \tag{14}$$

Felizmente, não é necessário gerar todas as restrições para resolver $(PQ)_{\pm t}^{\pi \mathcal{N}}$. De fato, é necessário apenas um subconjunto de restrições, que podem ser escolhidas através de um algoritmo gerador de restrições. Mostraremos a seguir, que a separação dessas restrições pode ser feita em tempo polinomial.

Para $\pi \in S_n$, $(N_0, N_1) \in \mathcal{N}$ e $x \in [0, 1]^n$ defina

$$\begin{aligned} \Theta(\pi, N_0, N_1; x) &= \sum_{i \in N} \sum_{\substack{\pi(j) \in N_1 \\ j < i}} \alpha_{\pi(j)\pi(i)} (\bar{x}_{\pi(i)} + \bar{x}_{\pi(j)}) + \sum_{i \in N} \sum_{\substack{\pi(j) \in N_0 \\ j < i}} \alpha_{\pi(j)\pi(i)} (x_{\pi(i)} + x_{\pi(j)}) \\ &+ \sum_{i \in N} \sum_{\substack{\pi(j) \in N_1 \\ j < i}} \beta_{\pi(j)\pi(i)} x_{\pi(i)} + \sum_{i \in N} \sum_{\substack{\pi(j) \in N_0 \\ j < i}} \beta_{\pi(j)\pi(i)} x_{\pi(j)}, \end{aligned}$$

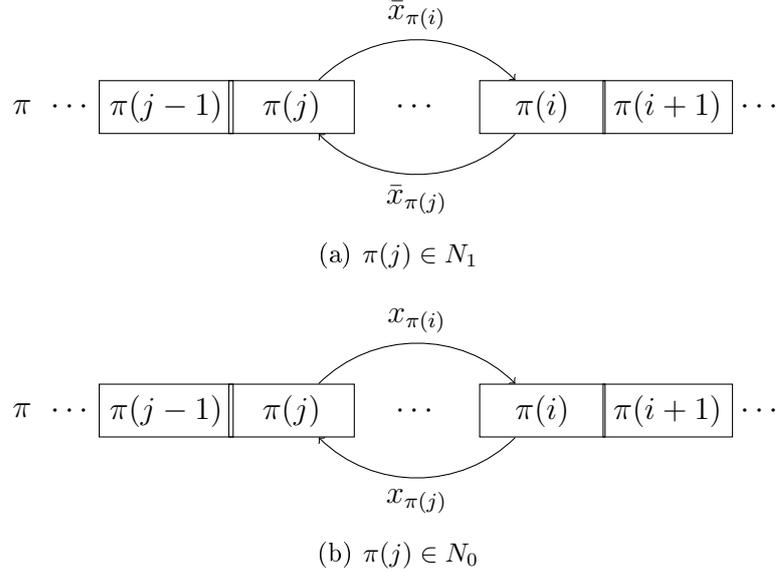
onde $\alpha_{\pi(j)\pi(i)} = \frac{-c_{\pi(j)\pi(i)}^-}{2} \geq 0$ e $\beta_{\pi(j)\pi(i)} = c_{\pi(j)\pi(i)}^+ \geq 0 \quad \forall i \in N \quad \forall j \in N \quad j < i$.

O problema de separação consiste em, dado $x^* \in [0, 1]^n$, resolver:

$$\min_{(N_0, N_1) \in \mathcal{N}} \min_{\pi \in S_n} \Theta(\pi, N_0, N_1; x^*) \tag{15}$$

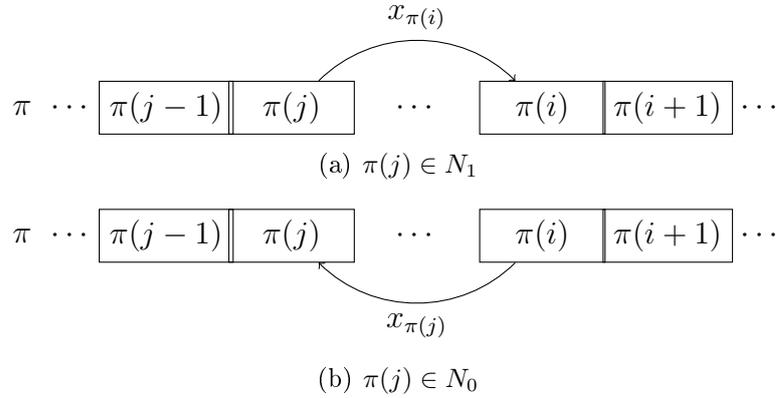
Primeiro mostraremos como encontrar a melhor permutação $\pi \in S_n$, fixada uma partição $(N_0, N_1) \in \mathcal{N}$. As Figuras 2 e 3 ilustram os coeficientes α e β que aparecem na expressão de $\Theta(\pi, N_0, N_1; x)$.

Figura 2: Coeficientes α que aparecem em $\Theta(\pi, N_0, N_1; x)$. As setas indicam a que variáveis o coeficiente $\alpha_{\pi(j)\pi(i)}$, $j < i$, está associado.



Fonte: Elaborado pelo autor.

Figura 3: Coeficientes β que aparecem em $\Theta(\pi, N_0, N_1; x)$. As setas indicam a que variáveis o coeficiente $\beta_{\pi(j)\pi(i)}$, $j < i$, está associado.



Fonte: Elaborado pelo autor.

Lema 2.1. Para $x^* \in [0, 1]^n$ e $(\hat{N}_0, \hat{N}_1) \in \mathcal{N}$, a permutação $\hat{\pi} \in S_n$ que minimiza $\Theta(\pi, \hat{N}_0, \hat{N}_1; x^*)$ é tal que

$$\hat{y}_{\hat{\pi}(1)} \geq \hat{y}_{\hat{\pi}(2)} \geq \dots \geq \hat{y}_{\hat{\pi}(n)}, \quad (16)$$

onde $\hat{y}_j = \begin{cases} x_j^*, & j \in N_1, \\ \bar{x}_j^*, & j \in N_0. \end{cases}$

Demonstração. Fixe $(\hat{N}_0, \hat{N}_1) \in \mathcal{N}$ e $x^* \in [0, 1]^n$. Seja $\pi \in S_n$ que minimiza $\Theta(\pi, \hat{N}_0, \hat{N}_1; x^*)$. Suponha que π não satisfaça (16). Seja, então, $l < n$ o máximo inteiro tal que as primeiras l posições de \hat{y} , segundo π , estão ordenadas de forma não crescente. Então:

$$\hat{y}_{\pi(1)} \geq \hat{y}_{\pi(2)} \geq \dots \geq \hat{y}_{\pi(l)} < \hat{y}_{\pi(l+1)}.$$

Seja também $k \leq l$ o menor inteiro satisfazendo $\hat{y}_{\pi(l+1)} \geq \hat{y}_{\pi(k)}$, levando a

$$\hat{y}_{\pi(1)} \geq \dots \geq \hat{y}_{\pi(k-1)} \geq \hat{y}_{\pi(l+1)} > \hat{y}_{\pi(k)} \geq \dots \geq \hat{y}_{\pi(l)}.$$

Defina uma nova permutação π' que ordena as primeiras $l+1$ entradas de \hat{y} , ou seja,

$$\pi'(i) = \begin{cases} \pi(i), & i = 1 \dots k-1 \text{ ou } i = l+2 \dots n, \\ \pi(l+1), & i = k, \\ \pi(i-1), & i = k+1 \dots l+1. \end{cases}$$

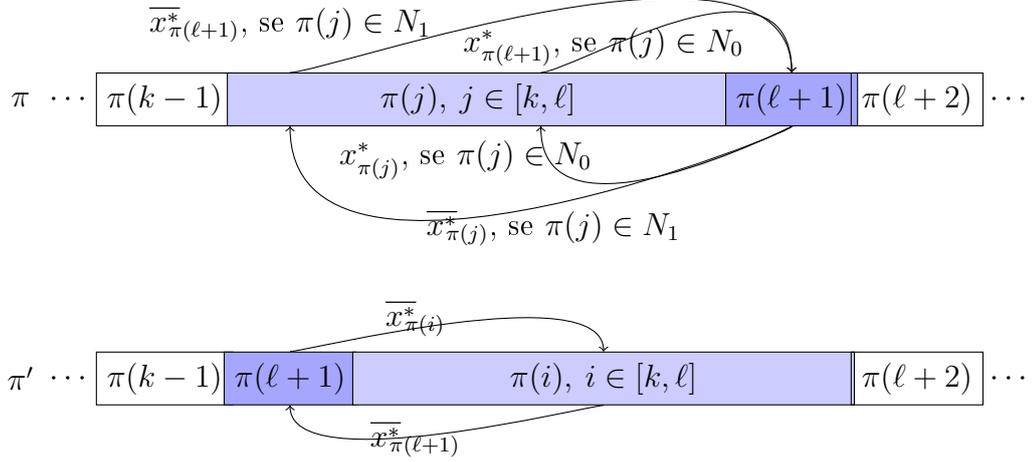
Seja $\Delta = \Theta(\pi', \hat{N}_0, \hat{N}_1; x^*) - \Theta(\pi, \hat{N}_0, \hat{N}_1; x^*)$. Primeiro suponha que $\pi(l+1) \in N_1$. Para facilitar a apresentação da expressão de Δ , a partir das figuras 2 e 3, construímos a Figura 4, que ilustra os coeficientes α e β que aparecem em $\Theta(\pi', \hat{N}_0, \hat{N}_1; x^*)$ e não em $\Theta(\pi, \hat{N}_0, \hat{N}_1; x^*)$, ou vice-versa.

Temos então que

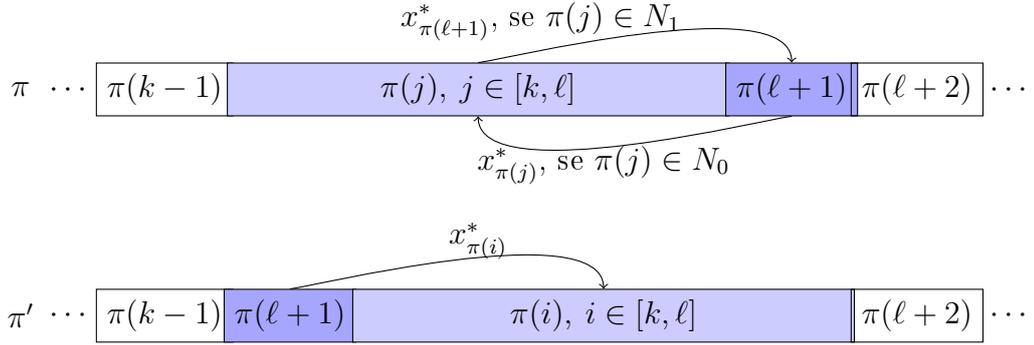
$$\begin{aligned} \Delta &= \sum_{i=k}^l \alpha_{\pi(l+1)\pi(i)} (\overline{x^*}_{\pi(i)} + \overline{x^*}_{\pi(l+1)}) - \sum_{\substack{j=k \\ \pi(j) \in N_1}}^l \alpha_{\pi(l+1)\pi(j)} (\overline{x^*}_{\pi(j)} + \overline{x^*}_{\pi(l+1)}) - \sum_{\substack{j=k \\ \pi(j) \in N_0}}^l \alpha_{\pi(l+1)\pi(j)} (x_{\pi(j)}^* + x_{\pi(l+1)}^*) \\ &+ \sum_{i=k}^l \beta_{\pi(l+1)\pi(i)} x_{\pi(i)}^* - \sum_{\substack{j=k \\ \pi(j) \in N_1}}^l \beta_{\pi(j)\pi(l+1)} x_{\pi(l+1)}^* - \sum_{\substack{j=k \\ \pi(j) \in N_0}}^l \beta_{\pi(j)\pi(l+1)} x_{\pi(j)}^* \\ &= \sum_{\substack{j=k \\ \pi(j) \in N_0}}^l \alpha_{\pi(l+1)\pi(i)} (\overline{x^*}_{\pi(l+1)} + \overline{x^*}_{\pi(j)} - x_{\pi(j)}^* - x_{\pi(l+1)}^*) + \sum_{\substack{j=k \\ \pi(j) \in N_1}}^l \beta_{\pi(l+1)\pi(i)} (x_{\pi(j)}^* - x_{\pi(l+1)}^*) \\ &= \sum_{\substack{j=k \\ \pi(j) \in N_0}}^l \alpha_{\pi(l+1)\pi(i)} 2(\underbrace{\overline{x^*}_{\pi(j)}}_{\in N_0} - \underbrace{x_{\pi(l+1)}^*}_{\in N_1}) + \sum_{\substack{j=k \\ \pi(j) \in N_1}}^l \beta_{\pi(l+1)\pi(i)} (\underbrace{x_{\pi(j)}^*}_{\in N_1} - \underbrace{x_{\pi(l+1)}^*}_{\in N_1}) \\ &= \sum_{\substack{j=k \\ \pi(j) \in N_0}}^l \underbrace{\alpha_{\pi(l+1)\pi(i)}}_{\geq 0} 2(\underbrace{\hat{y}_{\pi(j)}}_{< 0} - \underbrace{\hat{y}_{\pi(l+1)}}_{< 0}) + \sum_{\substack{j=k \\ \pi(j) \in N_1}}^l \underbrace{\beta_{\pi(l+1)\pi(i)}}_{\geq 0} (\underbrace{\hat{y}_{\pi(j)}}_{< 0} - \underbrace{\hat{y}_{\pi(l+1)}}_{< 0}) \\ &\leq 0, \end{aligned}$$

ou seja, $\Theta(\pi', \hat{N}_0, \hat{N}_1; x^*) \leq \Theta(\pi, \hat{N}_0, \hat{N}_1; x^*)$. Então π' também minimiza $\Theta(\pi, \hat{N}_0, \hat{N}_1; \hat{x})$ e ordena pelo menos mais uma entrada em \hat{y} . Repetindo o processo um número finito de passos, chegamos a uma permutação $\hat{\pi}$ que satisfaz (16).

Figura 4: Diferença entre $\Theta(\pi, N_0, N_1; \hat{x})$ e $\Theta(\pi', N_0, N_1; \hat{x})$ quando $\pi(l+1) \in N_1$. Em azul as posições diferentes entre π e π' . As setas indicam os coeficientes α (Fig. a) e β (Fig. b) que aparecem $\Theta(\pi, N_0, N_1; \hat{x})$ mas não em $\Theta(\pi', N_0, N_1; \hat{x})$ (figura superior) e vice-versa (figura inferior), apresentando as variáveis associadas.



(a) Coeficientes α . Ver Figura 2



(b) Coeficientes β . Ver Figura 3

Fonte: Elaborado pelo autor.

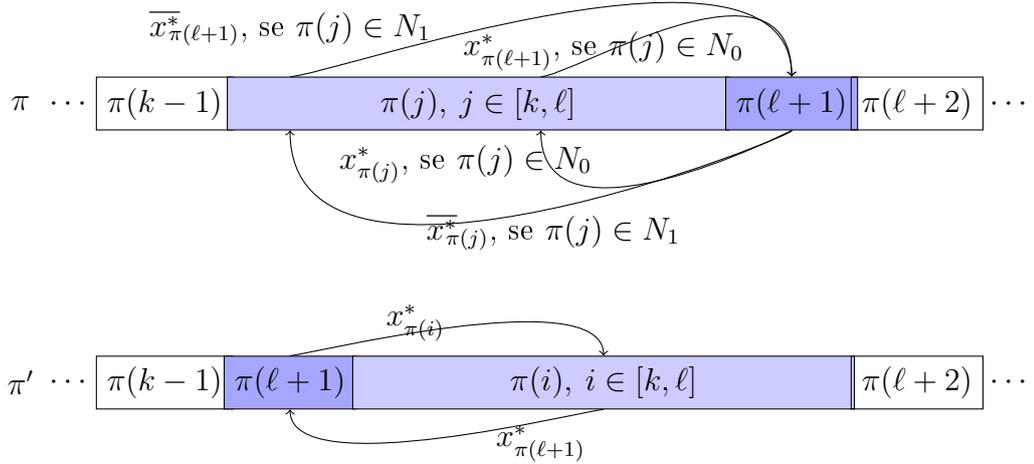
Agora suponha que $\pi(l+1) \in N_0$. Similarmente construímos a Figura 5 para ilustrar a expressão de Δ . Neste caso,

$$\begin{aligned} \Delta = & \sum_{i=k}^l \alpha_{\pi(l+1)\pi(i)} (x_{\pi(i)}^* + x_{\pi(l+1)}^*) - \sum_{\substack{j=k \\ \pi(j) \in N_1}}^l \alpha_{\pi(l+1)\pi(j)} (\bar{x}_{\pi(l+1)}^* + \bar{x}_{\pi(j)}^*) - \sum_{\substack{j=1 \\ \pi(j) \in N_0}}^l \alpha_{\pi(l+1)\pi(j)} (x_{\pi(l+1)}^* + x_{\pi(j)}^*) \\ & + \sum_{i=k}^l \beta_{\pi(l+1)\pi(i)} x_{\pi(l+1)}^* - \sum_{\substack{i=k \\ \pi(j) \in N_1}}^l \beta_{\pi(j)\pi(l+1)} x_{\pi(l+1)}^* - \sum_{\substack{j=k \\ \pi(j) \in N_0}}^l \beta_{\pi(l+1)\pi(j)} x_{\pi(j)}^* \end{aligned}$$

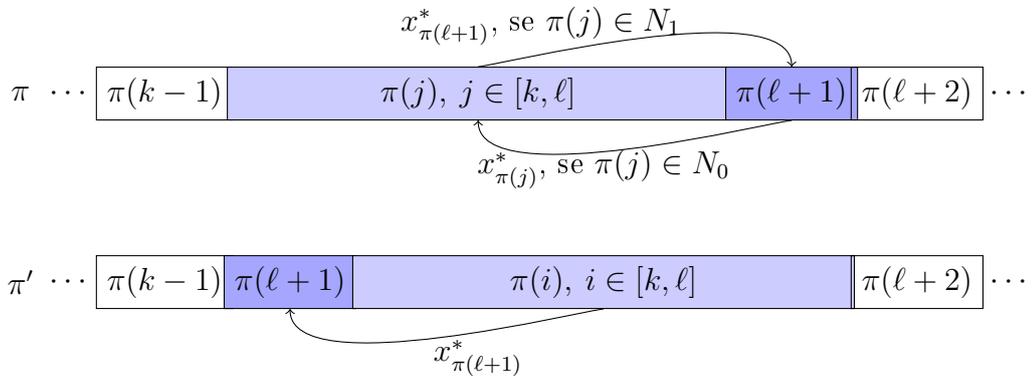
$$\begin{aligned}
&= \sum_{\substack{j=k \\ \pi(j) \in N_1}}^l \alpha_{\pi(l+1)\pi(j)} \left(x_{\pi(l+1)}^* + \underbrace{x_{\pi(j)}^*}_{\in N_1} - \underbrace{\bar{x}_{\pi(l+1)}}_{\in N_0} - \underbrace{\bar{x}_{\pi(j)}}_{\in N_0} \right) + \sum_{\substack{j=k \\ \pi(j) \in N_0}}^l \beta_{\pi(j)\pi(l+1)} \overbrace{\left(x_{\pi(l+1)}^* - x_{\pi(j)}^* \right)}^{\bar{x}_{\pi(j)}^* - \bar{x}_{\pi(l+1)}^*} \\
&= \sum_{\substack{j=k \\ \pi(j) \in N_1}}^l \underbrace{\alpha_{\pi(l+1)\pi(j)}}_{\geq 0} \underbrace{2(\hat{y}_{\pi(j)} - \hat{y}_{\pi(l+1)})}_{< 0} + \sum_{\substack{j=k \\ \pi(j) \in N_0}}^l \underbrace{\beta_{\pi(j)\pi(l+1)}}_{\geq 0} \underbrace{(\hat{y}_{\pi(j)} - \hat{y}_{\pi(l+1)})}_{< 0} \\
&\leq 0,
\end{aligned}$$

o que implica $\Theta(\pi', \hat{N}_0, \hat{N}_1; x^*) \leq \Theta(\pi, \hat{N}_0, \hat{N}_1; x^*)$. Então veja que π' também minimiza $\Theta(\pi, \hat{N}_0, \hat{N}_1; x^*)$ e ordena pelo menos mais uma entrada em \hat{y} . Repetindo o processo um número finito de passos, chegamos a uma permutação $\hat{\pi}$ que satisfaz (16). \square

Figura 5: Diferença entre $\Theta(\pi, N_0, N_1; \hat{x})$ e $\Theta(\pi', N_0, N_1; \hat{x})$ quando $\pi(l+1) \in N_0$. Em azul as posições diferentes entre π e π' . As setas indicam os coeficientes α (Fig. a) e β (Fig. b) que aparecem $\Theta(\pi, N_0, N_1; \hat{x})$ mas não em $\Theta(\pi', N_0, N_1; \hat{x})$ (figura superior) e vice-versa (figura inferior), apresentando as variáveis associadas.



(a) Coeficientes α . Ver Figura 2



(b) Coeficientes β . Ver Figura 3

Fonte: Elaborado pelo autor.

Teorema 2.6. Dado $x^* \in [0, 1]^n$, a solução do problema de separação (15) ocorre em (\hat{N}_0, \hat{N}_1) e $\hat{\pi} \in S_n$ tais que

$$\hat{N}_0 = \{j \in N : \overline{x_j^*} > x_j^*\}, \quad \hat{N}_1 = \{j \in N : x_j^* \geq \overline{x_j^*}\}, \quad (17)$$

$$\hat{y}_{\hat{\pi}(1)} \geq \hat{y}_{\hat{\pi}(2)} \geq \dots \geq \hat{y}_{\hat{\pi}(n)}, \quad (18)$$

onde $\hat{y}_j = \overline{x_j^*}$, se $j \in \hat{N}_0$, e $\hat{y}_j = x_j^*$, se $j \in \hat{N}_1$.

Demonstração. Seja $(N_0, N_1) \in \mathcal{N}$, $\pi \in S_n$ uma solução ótima para (15). Pelo Lema 2.1, podemos admitir que π é tal que

$$y_{\pi(1)} \geq y_{\pi(2)} \geq \dots \geq y_{\pi(n)} \text{ onde } y_j = \overline{x_j^*} \text{ se } j \in N_0, y_j = x_j^*, \text{ se } j \in N_1.$$

Se $(N_0, N_1) = (\hat{N}_0, \hat{N}_1)$, obtemos o resultado desejado. Caso contrário, existe um inteiro $l \geq 1$ tal que

$$\pi(j) \in N_0 \Leftrightarrow \pi(j) \in \hat{N}_0 \quad \forall j = 1, \dots, l-1 \quad \text{e}$$

$$\pi(l) \in N_0 \setminus \hat{N}_0 = N_0 \cap \hat{N}_1 \quad \text{ou} \quad \pi(l) \in N_1 \setminus \hat{N}_1 = N_1 \cap \hat{N}_0$$

Suponha primeiro que $\pi(l) \in N_0 \cap \hat{N}_1$. Defina $N'_0 = N_0 \setminus \{\pi(l)\}$ e $N'_1 = N_1 \cup \{\pi(l)\}$. Temos que $y_{\pi(l)} = \overline{x_{\pi(l)}^*} \leq x_{\pi(l)}^* = \overline{y_{\pi(l)}}$. Seja $k \leq l$ tal que

$$y_{\pi(1)} \geq \dots \geq y_{\pi(k-1)} \geq \overline{y_{\pi(l)}} \geq y_{\pi(k)} \geq \dots \geq y_{\pi(l-1)} \geq y_{\pi(l+1)} \geq \dots \geq y_{\pi(n)}.$$

Pelo Lema 2.1, a ordenação π' que minimiza $\Theta(\pi, N'_0, N'_1; x^*)$ é a que ordena os valores acima, ou seja,

$$\pi'(j) = \begin{cases} \pi(j), & j = 1 \dots k-1 \text{ ou } i = l+1 \dots n, \\ \pi(l), & j = k, \\ \pi(j-1), & j = k+1 \dots l. \end{cases} \quad (19)$$

Seja $W = \Theta(\pi', N'_0, N'_1; x^*) - \Theta(\pi, N_0, N_1; x^*)$. Então,

$$\begin{aligned}
W &= \sum_{i=1}^{l-1} \alpha_{\pi(l)\pi(i)}(\overline{x^*_{\pi(i)}} + \overline{x^*_{\pi(l)}}) - \sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l-1} \alpha_{\pi(l)\pi(j)}(\overline{x^*_{\pi(j)}} + \overline{x^*_{\pi(l)}}) - \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l-1} \alpha_{\pi(l)\pi(j)}(x^*_{\pi(j)} + x^*_{\pi(l)}), \\
&+ \sum_{i=k}^{l-1} \beta_{\pi(l)\pi(i)}x^*_{\pi(i)} - \sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l-1} \beta_{\pi(l)\pi(j)}x^*_{\pi(l)} - \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l-1} \beta_{\pi(j)\pi(l)}x^*_{\pi(j)} \\
&= \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l-1} \alpha_{\pi(l)\pi(j)}(\overline{x^*_{\pi(j)}} + \overline{x^*_{\pi(l)}} - x^*_{\pi(j)} - x^*_{\pi(l)}) + \sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l-1} \beta_{\pi(j)\pi(l)}(x^*_{\pi(j)} - x^*_{\pi(l)}) \\
&= \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l-1} \alpha_{\pi(j)\pi(l)}2(\underbrace{\overline{x^*_{\pi(j)}}}_{=y_{\pi(j)}} - \underbrace{x^*_{\pi(l)}}_{=\overline{y}_{\pi(l)}}) + \sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l-1} \beta_{\pi(j)\pi(l)}(\underbrace{x^*_{\pi(j)}}_{=y_{\pi(j)}} - \underbrace{x^*_{\pi(l)}}_{=\overline{y}_{\pi(l)}}),
\end{aligned}$$

levando a $\Theta(\pi', N'_0, N'_1; x^*) \leq \Theta(\pi, N_0, N_1; x^*)$. Logo $(N'_0, N'_1) \in \mathcal{N}$, $\pi' \in S_n$ também é solução ótima e (N'_0, N'_1) coincide em (\hat{N}_0, \hat{N}_1) em pelo menos mais um elemento (em relação a (N_0, N_1)). Repetindo o processo um número finito de passos, mostramos que (\hat{N}_0, \hat{N}_1) , $\hat{\pi} \in S_n$ é solução ótima.

Para o caso complementar, considere que $\pi(l) \in N_1 \cap \hat{N}_0$. De forma similar, vamos definir $N'_0 = N_0 \cup \{\pi(l)\}$ e $N'_1 = N_1 \setminus \{\pi(l)\}$. Agora, $y_{\pi(l)} = x^*_{\pi(l)} < \overline{x^*_{\pi(l)}} = \overline{y}_{\pi(l)}$. Seja $k \leq l$ tal que

$$y_{\pi(1)} \geq \dots \geq y_{\pi(k-1)} \geq \overline{y}_{\pi(l)} > y_{\pi(k)} > \dots \geq y_{\pi(l-1)} \geq y_{\pi(l+1)} \geq \dots \geq y_{\pi(n)}.$$

Pelo Lema 2.1, esta ordenação define a permutação π' , dada por (19), que minimiza $\Theta(\pi, N'_0, N'_1; x^*)$. Neste caso, $W = \Theta(\pi', N'_0, N'_1; x^*) - \Theta(\pi, N_0, N_1; x^*)$, onde

$$\begin{aligned}
W &= \sum_{i=k}^{l-1} \alpha_{\pi(l)\pi(i)}(x^*_{\pi(i)} + x^*_{\pi(l)}) - \sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l-1} \alpha_{\pi(l)\pi(j)}(\overline{x^*_{\pi(l)}} + \overline{x^*_{\pi(j)}}) - \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l-1} \alpha_{\pi(l)\pi(j)}(x^*_{\pi(l)} + x^*_{\pi(j)}) \\
&+ \sum_{i=k}^{l-1} \beta_{\pi(l)\pi(i)}x^*_{\pi(l)} - \sum_{\substack{i=k \\ \pi(j) \in N_1}}^{l-1} \beta_{\pi(l)\pi(j)}x^*_{\pi(l)} - \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l-1} \beta_{\pi(l)\pi(i)}x^*_{\pi(j)} \\
&= \sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l-1} \alpha_{\pi(l)\pi(j)}(x^*_{\pi(l)} + x^*_{\pi(j)} - \overline{x^*_{\pi(l)}} - \overline{x^*_{\pi(j)}}) + \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l-1} \beta_{\pi(j)\pi(l)}(x^*_{\pi(l)} - x^*_{\pi(j)}) \\
&= \sum_{\substack{j=k \\ \pi(j) \in N_1}}^{l-1} \alpha_{\pi(l)\pi(j)}2(\underbrace{\overline{x^*_{\pi(j)}}}_{=y_{\pi(j)}} - \underbrace{\overline{x^*_{\pi(l)}}}_{=\overline{y}_{\pi(l)}}) + \sum_{\substack{j=k \\ \pi(j) \in N_0}}^{l-1} \beta_{\pi(j)\pi(l)}(\underbrace{\overline{x^*_{\pi(j)}}}_{=y_{\pi(j)}} - \underbrace{\overline{x^*_{\pi(l)}}}_{=\overline{y}_{\pi(l)}}),
\end{aligned}$$

ou ainda $\Theta(\pi', N'_0, N'_1; x^*) \leq \Theta(\pi, N'_0, N'_1; x^*)$. Novamente, obtemos uma solução ótima

(N'_0, N'_1) , $\pi' \in S_n$ onde (N'_0, N'_1) coincide com (\hat{N}_0, \hat{N}_1) em pelo menos mais um elemento, levando ao resultado desejado. \square

Apresentamos a seguir, para efeitos didáticos, um exemplo do funcionamento da t -linearização para coeficientes arbitrários. Considere o seguinte problema:

$$\begin{aligned}
 (PQ)_{ex} \quad \max \quad & -5x_1 - 2x_2 + 14x_3 + 7x_4 + 3x_5 \\
 & +7x_1x_2 - 4x_1x_3 - 7x_1x_4 + 9x_1x_5 \\
 & -5x_2x_3 - 3x_2x_4 + 3x_2x_5 \\
 & -x_3x_4 - 4x_3x_5 \\
 & +8x_4x_5 \\
 \text{s.a} \quad & x_i \in \{0, 1\} \quad i = 1, \dots, 5.
 \end{aligned}$$

No modelo t -linearizado (14), a função objetivo torna-se

$$\max t + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n c_{ij}^- x_i + \sum_{i=1}^n b_i x_i = t - 10.5x_1 - 6x_2 + 7x_3 + 1.5x_4 + x_5 \quad (20)$$

Vamos agora apresentar a geração de uma restrição de (14), ou seja, do conjunto:

$$\begin{aligned}
 t \leq & \sum_{\substack{i \in N\pi(j) \in N_1 \\ j < i}} c_{\pi(j)\pi(i)}^+ x_{\pi(i)} + \sum_{\substack{i \in N\pi(j) \in N_0 \\ j < i}} c_{\pi(j)\pi(i)}^+ x_{\pi(j)} + \frac{1}{2} \sum_{\substack{i \in N \\ \pi(j) \in N_1 \\ j < i}} -c_{\pi(j)\pi(i)}^- (\bar{x}_{\pi(i)} + \bar{x}_{\pi(j)}) \\
 & + \frac{1}{2} \sum_{\substack{i \in N\pi(j) \in N_0 \\ j < i}} -c_{\pi(j)\pi(i)}^- (x_{\pi(i)} + x_{\pi(j)}) \quad \forall \pi \in S_n \quad \forall (N_0, N_1) \in \mathcal{N},
 \end{aligned} \quad (21)$$

a partir de uma solução relaxada $x \in [0, 1]^n$ qualquer. Tomemos, por exemplo, $x = (0.3, 0.3, 1, 0.9, 0.6)$. Temos $\bar{x} = (0.7, 0.7, 0, 0.1, 0.4)$. Seguindo o Teorema 2.6, $N_0 = \{1, 2\}$, $N_1 = \{3, 4, 5\}$ e $y = (0.7, 0.7, 1, 0.9, 0.6)$, levando à permutação $\pi = (3, 4, 2, 1, 5)$ que ordena as componentes de y em ordem não decrescente (ou $\pi = (3, 4, 1, 2, 5)$, devido ao empate entre as duas primeiras componentes de y). Note que essa mesma permutação seria obtida a partir de $x = (0, 0, 1, 1, 1)$, a solução que maximiza a expressão em x de (20).

Com a permutação $\pi = (3, 4, 2, 1, 5)$, os termos da restrição (21) ficam:

$$\begin{aligned}
\sum_{i \in N} \sum_{\substack{j \in N_1 \\ j < i}} c_{\pi(j)\pi(i)}^+ x_{\pi(i)} &= c_{45}^+ x_5 = 8x_5, \\
\sum_{i \in N} \sum_{\substack{j \in N_0 \\ j < i}} c_{\pi(j)\pi(i)}^+ x_{\pi(j)} &= (c_{21}^+ x_2 + c_{25}^+ x_2 + c_{15}^+ x_1) = 10x_2 + 9x_1, \\
\frac{1}{2} \sum_{i \in N} \sum_{\substack{j \in N_1 \\ j < i}} -c_{\pi(j)\pi(i)}^- (\bar{x}_{\pi(i)} + \bar{x}_{\pi(j)}) &= -c_{34}^- (2 - x_4 - x_3) - c_{32}^- (2 - x_2 - x_3) - c_{42}^- (2 - x_2 - x_4) \\
&\quad - c_{31}^- (2 - x_1 - x_3) - c_{41}^- (2 - x_1 - x_4) - c_{35}^- (2 - x_5 - x_3) \\
&= -5.5x_1 - 4x_2 - 7x_3 - 5.5x_4 - 2x_5 + 24, \\
\frac{1}{2} \sum_{i \in N} \sum_{\substack{j \in N_0 \\ j < i}} -c_{\pi(j)\pi(i)}^- (x_{\pi(i)} + x_{\pi(j)}) &= 0.
\end{aligned}$$

A restrição obtida é, portanto, $t \leq 3.5x_1 + 6x_2 - 7x_3 - 5.5x_4 + 6x_5 + 24$.

Podemos adicioná-la então ao modelo $(PQ)_{ex}$, que se torna:

$$\begin{aligned}
(PQ)_{ex} \quad \max \quad & t - 10.5x_1 - 6x_2 + 7x_3 + 1.5x_4 + x_5 \\
s.a \quad & t \leq 3.5x_1 + 6x_2 - 7x_3 - 5.5x_4 + 6x_5 + 24 \\
& 0 \leq t, \quad x_i \in \{0, 1\} \quad i = 1, \dots, 5.
\end{aligned}$$

A seguir mostramos que o limite superior fornecido pela t -linearização é menor ou igual àquele da relaxação clássica. Os experimentos computacionais do Capítulo 4 mostram que essa superioridade pode ser estrita.

Teorema 2.7. *O valor da relaxação de (2) é pelo menos aquele da relaxação de (14).*

Demonstração. Seja (x^*, t^*) uma solução ótima da relaxação linear de (14). Note que t^* é o valor ótimo do problema (15). Então, pelo Teorema 2.6, $t^* = \Theta(\hat{\pi}, \hat{N}_0, \hat{N}_1; x^*)$, onde a partição (\hat{N}_0, \hat{N}_1) e a permutação $\hat{\pi}$ satisfazem (17) e (18). Sem perda de generalidade, vamos supor que $\hat{\pi}$ é a permutação identidade (do contrário podemos renomear as variáveis). Assim,

$$\hat{N}_0 = \{j \in N : \bar{x}_j^* > x_j^*\}, \quad \hat{N}_1 = \{j \in N : x_j^* \geq \bar{x}_j^*\},$$

$$\max\{\bar{x}_j^*, x_j^*\} \geq \max\{\bar{x}_i^*, x_i^*\} \quad \forall 1 \leq j < i \leq n.$$

Além disso,

$$t^* = \sum_{i \in N} \sum_{\substack{j \in \hat{N}_1 \\ j < i}} c_{ji}^+ x_i^* + \sum_{i \in N} \sum_{\substack{j \in \hat{N}_0 \\ j < i}} c_{ji}^+ x_j^* + \sum_{i \in N} \sum_{\substack{j \in \hat{N}_1 \\ j < i}} -\frac{1}{2} c_{ji}^- (\bar{x}_i^* + \bar{x}_j^*) + \sum_{i \in N} \sum_{\substack{j \in \hat{N}_0 \\ j < i}} -\frac{1}{2} c_{ji}^- (x_i^* + x_j^*).$$

Defina $y^* = [y_{ji}^*]_{1 \leq j < i \leq n}$ tal que $y_{ji}^* = \max\{0, x_i^* + x_j^* - 1\}$, se $c_{ji} < 0$, e $y_{ji}^* = \min\{x_i^*, x_j^*\}$,

se $c_{ji} \geq 0$. Como $x^* \in [0, 1]^n$, temos que

$$0 \leq \max\{0, x_i^* + x_j^* - 1\} \leq y_{ji}^* \leq \min\{x_i^*, x_j^*\} \leq 1, \quad 1 \leq j < i \leq n.$$

Logo, y^* é viável para a relaxação linear de (2). Observe também os valores individuais de y_{ji}^* , conforme os casos da Tabela 1:

Tabela 1 – Valores individuais de y_{ji}^*

$j < i$	$c_{ji} < 0$	$c_{ji} \geq 0$
$j \in \hat{N}_0$	$y_{ji}^* = 0$	$y_{ji}^* = x_j^*$
$j \in \hat{N}_1$	$y_{ji}^* = x_i^* + x_j^* - 1$	$y_{ji}^* = x_i^*$

Fonte: Elaborado pelo autor.

Consequentemente, o valor da função objetivo de (2) em y^* é $Q + \sum_{i=1}^n b_i x_i^*$,

onde

$$\begin{aligned}
Q &= \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji} y_{ji}^* \\
&= \sum_{i \in N} \sum_{\substack{j \in \hat{N}_1 \\ j < i}} c_{ji}^+ y_{ji}^* + \sum_{i \in N} \sum_{\substack{j \in \hat{N}_0 \\ j < i}} c_{ji}^+ y_{ji}^* + \sum_{i \in N} \sum_{\substack{j \in \hat{N}_1 \\ j < i}} c_{ji}^- y_{ji}^* + \sum_{i \in N} \sum_{\substack{j \in \hat{N}_0 \\ j < i}} c_{ji}^- y_{ji}^* \\
&= \sum_{i \in N} \sum_{\substack{j \in \hat{N}_1 \\ j < i}} c_{ji}^+ x_i^* + \sum_{i \in N} \sum_{\substack{j \in \hat{N}_0 \\ j < i}} c_{ji}^+ x_j^* + \sum_{i \in N} \sum_{\substack{j \in \hat{N}_1 \\ j < i}} c_{ji}^- (x_j^* + x_i^* - 1) \\
&= \sum_{i \in N} \sum_{\substack{j \in \hat{N}_1 \\ j < i}} c_{ji}^+ x_i^* + \sum_{i \in N} \sum_{\substack{j \in \hat{N}_0 \\ j < i}} c_{ji}^+ x_j^* + \sum_{i \in N} \sum_{\substack{j \in \hat{N}_1 \\ j < i}} -\frac{1}{2} c_{ji}^- (\bar{x}_i^* + \bar{x}_j^*) + \sum_{i \in N} \sum_{\substack{j \in \hat{N}_1 \\ j < i}} \frac{1}{2} c_{ji}^- (x_i^* + x_j^*) \\
&= t^* + \sum_{i \in N} \sum_{\substack{j \in N \\ j < i}} \frac{1}{2} c_{ji}^- (x_i^* + x_j^*) \\
&= t^* + \sum_{i \in N} \sum_{j \in N} \frac{1}{2} c_{ij}^- x_i^*
\end{aligned}$$

Comparando então os valores das funções objetivo de (2) em (x^*, y^*) e de (14) em (x^*, t^*) , concluímos que eles são iguais. Portanto, o valor ótimo de (2) é pelo menos o valor de (x^*, y^*) , e o resultado segue. \square

2.4 Suavização Hiperbólica

A técnica da suavização hiperbólica foi apresentada em Santos (1997) para resolver problemas de otimização com funções não diferenciáveis. Ela engloba o uso de duas abordagens, quais sejam, a suavização de funções não diferenciáveis e a penalização hiperbólica, esta última originalmente introduzida por Xavier (1982). A primeira abordagem consiste em substituir uma função não diferenciável por uma família de funções diferenciáveis que se aproximem gradativamente da função original. Já a característica comum aos mé-

todos de penalidades é a transformação de um problema de otimização com restrições num problema sem restrições, cuja função objetivo agrega um termo para penalizar soluções inviáveis (Lavor, 2004). Normalmente, esse termo é ponderado por um parâmetro de penalidade, gerando assim um problema irrestrito paramétrico, que vai se tornando equivalente ao original, à medida que o parâmetro cresce.

Combinando as duas abordagens, a suavização hiperbólica procura resolver um problema não diferenciável, transformando-o primeiramente num problema sem restrições (caso elas existam) e então resolvendo uma sequência de problemas diferenciáveis associados à família de funções definidas pela suavização.

Para ilustrar a suavização, iremos utilizar uma função básica que é fortemente usada nos trabalhos de Xavier e Xavier (2016) e Souza *et al.* (2011). Seja $f(u) = |u|$ a função valor absoluto, onde $u \in \mathbb{R}$. Para $\gamma > 0$, vamos definir

$$\theta(u, \gamma) = \sqrt{u^2 + \gamma^2}.$$

Note que a função θ tem as seguintes propriedades:

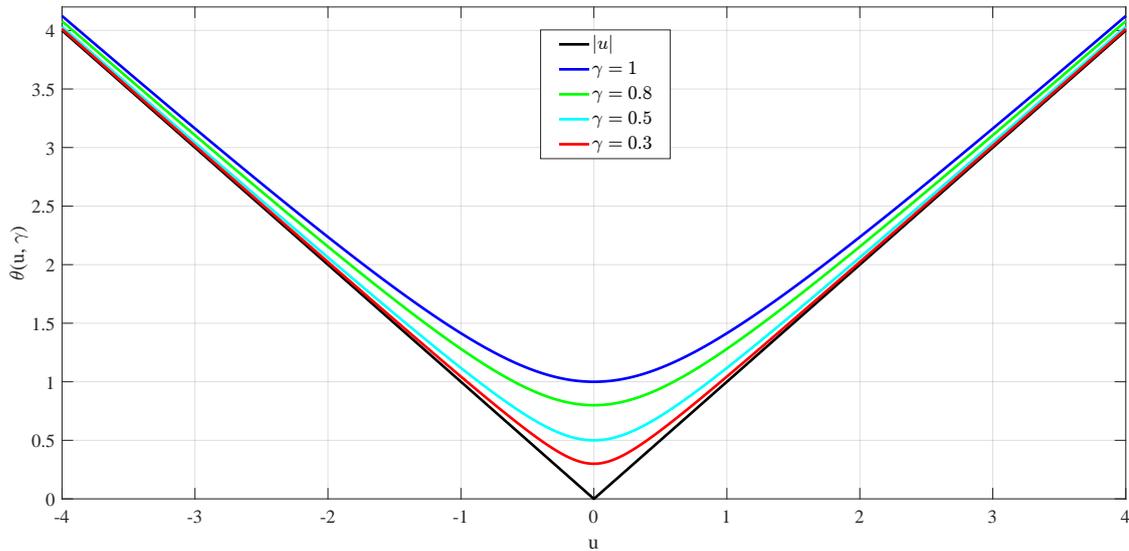
1. $\lim_{\gamma \rightarrow 0} \theta(u, \gamma) = |u|$;
2. θ é uma função C^∞ ;
3. $\theta'(u, \gamma) = u/(u^2 + \gamma^2)^{1/2}$;
4. $\theta''(u, \gamma) = \gamma^2/(u^2 + \gamma^2)^{3/2}$;
5. $\theta''(0, \gamma) = 1/\gamma$;
6. $\lim_{\gamma \rightarrow 0} \theta''(0, \gamma) = \infty$.

A propriedade 1 indica que podemos substituir a minimização de $f(u)$ pela minimização de $\theta(u, \gamma)$ para valores de γ gradativamente menores. A Figura 6 mostra de forma gradual como a função $\theta(u, \gamma)$ se aproxima da função $|u|$, à medida que $\gamma \rightarrow 0$. Como θ é infinitamente diferenciável (propriedade 2), podemos usar, para sua minimização, métodos de programação diferenciável, que demandam, por exemplo, informações das derivadas de primeira e segunda ordens, facilmente calculadas neste caso (propriedades 3 e 4). Em particular, a derivada de segunda ordem no ponto $u = 0$ (de não diferenciabilidade de f) tende a infinito quando $\gamma \rightarrow 0$.

De modo a descrever a técnica de penalidade, considere o problema P de programação matemática descrito a seguir:

$$\begin{aligned} (P) \quad & \min f(x) \\ & \text{s.a. } h_i(x) \leq 0, \quad i = 1, \dots, m, \\ & x \in \mathbb{R}^n, \end{aligned}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$, são funções quaisquer. A estratégia é transformar o problema P em um problema irrestrito \bar{P} , movendo para a função objetivo

Figura 6: $|u|$ e $\theta(u, \gamma)$, para $\gamma = [1, 0.8, 0.5, 0.3]$ 

Fonte: Adaptado de (Xavier e Xavier, 2016)

as restrições $h_i(x)$, através de penalidades:

$$(\bar{P}) \quad \min \quad f(x) + \sum_{i=1}^m g(h_i(x), \lambda)$$

onde $\lambda \geq 0$ e $g : \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ são, respectivamente, o parâmetro e a função de penalidade. Normalmente, temos $g(y, \lambda) = 0$ quando $y \leq 0$ e, quando $y > 0$, $g(y, \lambda) \rightarrow +\infty$ à medida que λ cresce. Vale destacar que o parâmetro de penalidade não precisa ser o mesmo para toda restrição. Em outros termos, podemos substituir λ por λ_i em \bar{P} .

As funções de penalidade clássicas são muitas vezes não diferenciáveis. Entre os exemplos típicos, podemos citar $g(y, \lambda) = \lambda \max\{0, y\}$. Sendo não diferenciável, a função de penalidade pode ser suavizada. Por exemplo, a função $g(y, \lambda) = \lambda \max\{0, y\}$ pode ser aproximada pela função suave

$$\phi(\lambda, \tau, y) = (\lambda y + \sqrt{\lambda^2 y^2 + \tau^2})/2,$$

onde $\tau \geq 0$ é um novo parâmetro. Note que, quando $\tau \rightarrow 0$, temos

$$\phi(\lambda, 0, y) = \begin{cases} 0, & \text{se } y \leq 0, \\ \lambda y, & \text{se } y > 0. \end{cases}$$

A função ϕ é chamada de suavização hiperbólica e possui as seguintes propriedades:

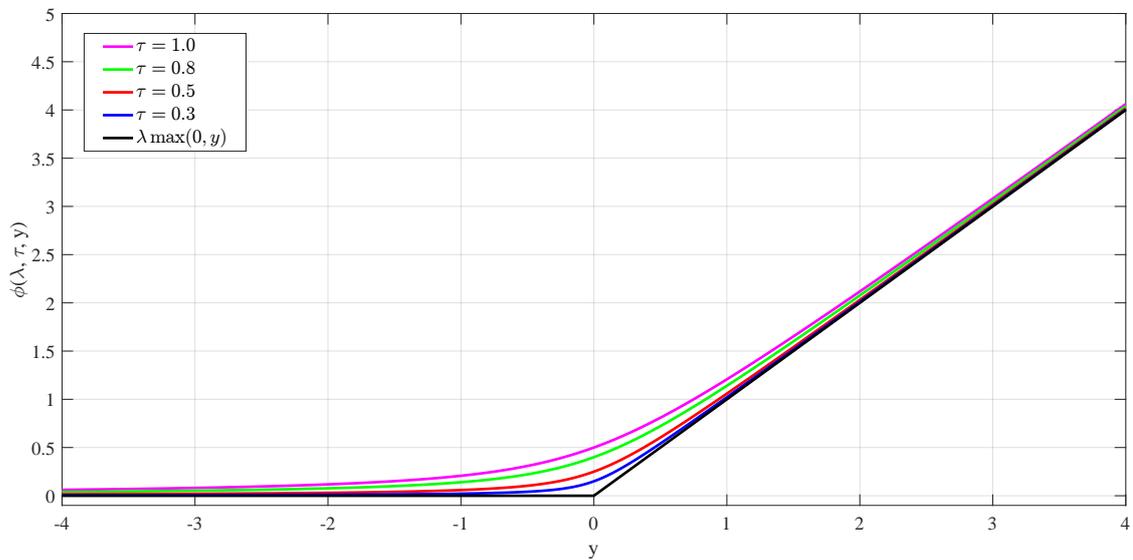
1. $\phi(\lambda, \tau, y) > g(y, \lambda)$, $\forall \tau > 0$;
2. $\lim_{\tau \rightarrow 0} \phi(\lambda, \tau, y) = g(y, \lambda)$;
3. $\phi(\lambda, \tau, y)$ é função C^∞ , crescente em y ;

4. $\phi'(\lambda, \tau, y) = \lambda + \lambda^2 y^2 / (\lambda^2 y^2 + \tau^2)^{1/2}$;
5. $\phi''(\lambda, \tau, y) = \lambda^2 \tau^2 / (\lambda^2 y^2 + \tau^2)^{3/2}$;
6. $\phi''(\lambda, \tau, 0) = \lambda^2 / \tau$;
7. $\lim_{\tau \rightarrow 0} \phi''(\lambda, \tau, 0) \rightarrow \infty$

Note pela propriedade 3 que, sendo ϕ crescente em y , o termo de penalidade aumenta com o descumprimento das restrições. Além disso, esse termo cresce quando o parâmetro de penalidade cresce.

A Figura 7 mostra de forma gradual como a função $\phi(\lambda, \tau, y)$ se aproxima da função $g(y, \lambda)$ à medida que $\tau \rightarrow 0$.

Figura 7: $g(y_i)$ e $\phi(\lambda, \tau, y_i)$, para $\tau = [1, 0.8, 0.5, 0.3]$



Fonte: Adaptado de (Souza *et al.*, 2011)

Dessa maneira, combinando suavização e penalização, a suavização hiperbólica trata o problema P através do problema

$$(\hat{P}) \quad \min \quad f(x) + \sum_{i=1}^m \phi(\lambda, \tau, h_i(x)).$$

E caso f seja não diferenciável, substitui também esta função por uma suavização.

3 CORTE MÁXIMO (MAX-CUT)

O problema do Corte Máximo (Max-Cut) é um dos problemas de otimização combinatória que, devido a sua dificuldade de solução exata e a seu domínio de aplicações, tais como a física estatística, fluxos em redes, circuitos VLSI etc, tem atraído o interesse científico de pesquisadores das áreas de matemática discreta e otimização matemática. Entre as muitas referências para o problema, citamos Deza e Laurent (2009), Benson, Ye, e Zhang (2000), Burer, Monteiro, e Zhang (2002) e Helmberg *et al.* (1996), que possuem maior relação com nosso trabalho. Sendo um problema NP - Difícil (Karp, 1972), é considerado um desafio computacional resolvê-lo de forma ótima, mesmo no caso de instâncias de tamanhos moderados (Krislock, Malick, e Rouoin, 2014). No entanto, vários casos especiais com resolução polinomial têm sido identificados (Orlova, 1972; Hadlock, 1975; Barahona, 1983).

O Max-Cut possui muitas formulações equivalentes, as quais aparecem com diferentes nomes na literatura. Além de corte máximo, as denominações mais comuns são *Maximum-2-Satisfiability*, *Weighted Signed Graph Balancing*, *Unconstrained Quadratic 0 – 1 Programming (UQP)* etc (Boros e Hammer, 1991).

Esse problema pode ser descrito da seguinte forma. É dado um grafo não direcionado $G = (V, E)$, com $|V| = n$ vértices e $|E| = m$ arestas ponderadas. Consideramos o peso $c_{ij} = c_{ji}$ para cada aresta $\{i, j\} \in E$, $c_{ij} = 0$ para $\{i, j\} \notin E$, e $c_{ii} = 0$, para todo $1 \leq i \leq n$. Qualquer bipartição $(S, \bar{S} := V \setminus S)$ dos vértices de V define um corte de G , qual seja, o subconjunto de arestas com uma extremidade em S e a outra em \bar{S} , onde S ou \bar{S} pode ser vazio. O problema do Max-Cut consiste em encontrar (S, \bar{S}) tal que a soma dos pesos das arestas do corte seja máximo. Denotamos por $c(S : \bar{S})$ o peso do corte, ou seja,

$$c(S : \bar{S}) = \sum_{(i,j) \in (S:\bar{S})} c_{ij}. \quad (22)$$

As arestas do corte definidas por (S, \bar{S}) são também denotadas por $\delta(S)$ ou, igualmente, $\delta(\bar{S})$. É importante destacar que, tomando $c_{ij} = 0$ para $\{i, j\} \notin E$, podemos sempre admitir que o grafo G é completo.

Para resolver o problema do Max-Cut, muitos métodos foram surgindo ao longo dos anos. Um survey de técnicas desenvolvidas antes de 1980 pode ser encontrado em Hansen (1979). Na década de 1980, buscava-se encontrar soluções exatas para o Max-Cut através de combinatória poliédrica. Essa abordagem obtinha bons resultados para instâncias esparsas, no entanto, para instâncias densas com mais de 50 nós, a abordagem não se mostrava satisfatória (Rendl, Rinaldi, e Wiegele, 2007). Goemans e Williamson (1995), usando relaxação e formulação de programação semidefinida, aumentaram a esperança de se resolver instâncias densas, ao mostrarem um algoritmo aproximativo que sempre oferece uma solução de valor pelo menos 0.87856 do ótimo, desde que os pesos das

arestas sejam não negativos.

Anos mais tarde, em Rendl, Rinaldi, e Wiegele (2007), os autores apresentaram um método exato, chamado Biq Mac - *Binary quadratic and Max cut Solver*, para encontrar soluções exatas para o problema do Max-Cut. Esse método usa relaxação de programação semidefinida combinada com desigualdades triangulares, que é resolvida pelo método de feixes. Os autores basearam essa abordagem em Fischer *et al.* (2006) e utilizaram dualidade lagrangiana para obter limites superiores com esforço computacional razoável. Os resultados mostraram que o Biq Mac dominou quase todos os métodos da época. Em Rendl, Rinaldi, e Wiegele (2010), os autores deram uma visão global dos métodos mais relevantes entre os anos 2000 e 2010 para resolver o Max-Cut de forma exata, além de melhorarem o Biq Mac, usando uma configuração de *Branch and Bound* que aplica uma versão dinâmica do método de feixes como um procedimento para obter limites. Dessa forma, conseguiram obter soluções exatas em tempo razoável para instâncias de até 100 nós, independente da densidade.

Em um trabalho mais recente, Krislock, Malick, e Rouoin (2014) apresentaram um algoritmo melhorado para encontrar soluções exatas para o Max-Cut. O algoritmo implementa o método *branch-(and-cut)-and bound*, usando inequações válidas e limites da programação semidefinida. Os resultados obtidos mostraram que tal algoritmo é mais rápido e robusto do que o Biq Mac, além de resolver em torno de 75% dos problemas de Biq Mac Library (Wiegele, 2007) em menos tempo que o Biq Mac.

3.1 Formulações Matemáticas

Apresentaremos nesta seção as formulações de programação matemática usuais para o Max-Cut. Como veremos a seguir, são modelos de programação linear ou quadrática, que podem ser convertidos um no outro.

3.1.1 Formulação por restrições triangulares

Seja $G = (V, E)$ um grafo. Se $F \subseteq E$ é um subconjunto de arestas de E , então o vetor $z^F \in \{0, 1\}^{|E|}$ tal que $z^F(e) = 1$ se, e somente se, $e \in F$ é chamado de vetor de incidência de F (Ben-Ameur, Mahjoub, e Neto, 2006). Para uma aresta $e = \{i, j\}$, vamos usar $z(e) = z_e = z_{ij} = z_{ji}$. Seja $P_c(G)$ a envoltória convexa dos vetores de incidência dos cortes de G , ou seja:

$$P_c(G) = \text{conv}\{z^F : F \text{ é um corte de } G\}.$$

$P_c(G)$ é chamado de polítopo de corte (*cut polytope*) de G . O problema do corte máximo

em G é, portanto, equivalente ao problema de programação linear

$$\max\{c^T z : z \in P_c(G)\}.$$

Sejam C o conjunto de arestas de um ciclo de G , $F \subseteq C$ tal que $|F|$ é ímpar e $W \subset V$. Seja $\delta(W)$ um corte de G . Uma vez que $\delta(W)$ contém um número par de arestas de C , se $F \subset \delta(W)$, então $\delta(W) \cap (C \setminus F) \neq \emptyset$. Consequentemente, as seguintes restrições, introduzidas em Barahona e Mahjoub (1986), são válidas para o politopo $P_c(G)$:

$$z(F) - z(C \setminus F) \leq |F| - 1, \quad \forall \text{ ciclo } C \text{ de } G, \forall F \subseteq C, |F| \text{ ímpar}, \quad (23)$$

$$0 \leq z(e) \leq 1, \quad \forall e \in E, \quad (24)$$

onde, para um conjunto de arestas S e variáveis z correspondentes, $z(S)$ denota o somatório $\sum_{e \in S} z_e$. As restrições (23) são chamadas de inequações de ciclo (*cycle inequalities*), e (24) são chamadas de inequações triviais (*trivial inequalities*) (Bonato *et al.*, 2014).

Quando C é um triângulo e, portanto, $|F| = 3$ ou $|F| = 1$, as desigualdades de ciclos tornam-se:

$$\begin{aligned} z_{ij} + z_{ik} + z_{jk} &\leq 2, \quad \forall \{i, j, k\} \subseteq V \\ z_{ij} - z_{ik} - z_{jk} &\leq 0, \quad \forall \{i, j, k\} \subseteq V \end{aligned} \quad (25)$$

Neste caso, passam a ser chamadas desigualdades triangulares (*triangular inequalities*) (Grötschel e Wakabayashi, 1989). Note que cada tripla $\{i, j, k\}$ define quatro restrições em (25). Observe ainda que essas desigualdades implicam (24). De fato, para $e = \{i, j\}$, trivialmente $z_{ij} + z_{ik} + z_{jk} \leq 2$ domina $z_e \leq 1$, enquanto a soma de $z_{ik} - z_{kj} - z_{ij} \leq 0$ com $z_{kj} - z_{ki} - z_{ij} \leq 2$ leva a $z_e \geq 0$.

O politopo definido pelas inequações (23) e (24) é chamado de *semimétrico*, enquanto o politopo dado pelas inequações triangulares é chamado de *métrico* (Ben-Ameur, Mahjoub, e Neto, 2006; Bonato *et al.*, 2014).

Quando o grafo é completo, podemos verificar que as desigualdades triangulares (e a integralidade das variáveis) são suficientes para definir uma formulação para o Max-Cut. De fato, nesse caso cada três vértices definem um triângulo e um subconjunto de arestas é um corte se, e somente se, ele contém 0 ou 2 arestas do triângulo. Esse é exatamente o sentido das desigualdades (25). Dessa forma, fazendo $c_{ij} = 0$ sempre que $\{i, j\} \notin E$, podemos encontrar o corte máximo de um grafo, resolvendo o seguinte problema de programação linear inteira:

$$\begin{aligned} (MC_z) \quad \max \quad & \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} z_{ij} \\ \text{s.a} \quad & z_{ij} + z_{ik} + z_{jk} \leq 2, \quad \forall \{i, j, k\} \subseteq V, \\ & z_{ij} - z_{ik} - z_{jk} \leq 0, \quad \forall \{i, j, k\} \subseteq V, \\ & z_{ij} \in \{0, 1\}. \end{aligned} \quad (26)$$

Note que, em (26), podemos considerar $i < j < k$ no primeiro grupo de restrições e $j < k$ no segundo.

Dado um ciclo C , uma *corda* de C é uma aresta cujas extremidades estão em C e são vértices não consecutivos ao atravessar C . O Teorema 3.1 a seguir, cuja demonstração pode ser encontrada em Barahona e Mahjoub (1986), fornece condições necessárias e suficientes para que as restrições (24) e (23) definam faceta.

Teorema 3.1. *As seguintes afirmações são verdadeiras:*

1. A inequação (23) define faceta de $P_c(G)$ se, e somente se, C não possui uma corda.
2. A inequação $z(e) \geq 0$ ($z(e) \leq 1$) define faceta de $P_c(G)$ se, e somente se, e não pertence a um triângulo.

Uma vez que as restrições (23) e (24) modelam o problema do corte máximo via programação linear inteira e, pelo Teorema 3.1, podem definir facetar, será útil um algoritmo de separação em tempo polinomial para essas restrições. Esse algoritmo pode ser encontrado em Barahona e Mahjoub (1986) e Ben-Ameur, Mahjoub, e Neto (2006).

3.1.2 Formulação binária $\{0, 1\}$

Se representarmos a bipartição (S, \bar{S}) pelo vetor binário $x \in \{0, 1\}^n$, com $x_i = 1$ se $i \in S$ e $x_i = 0$ se $i \in \bar{S}$, então o problema do Max-Cut pode ser formulado conforme a seguir, como um problema quadrático binário sem restrições

$$(MC_{\{0,1\}}) \quad \max \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}(x_i + x_j - 2x_i x_j) \quad (27)$$

s.a $x \in \{0, 1\}^n$.

A Tabela 2 ilustra a adequação da função objetivo em (27).

Tabela 2: Peso do corte da partição (S, \bar{S}) representado pelo vetor binário $x \in \{0, 1\}^n$

		Função Objetivo
x_i	x_j	$c_{ij}(x_i + x_j - 2x_i x_j)$
0	0	0
0	1	c_{ij}
1	0	c_{ij}
1	1	0

Fonte: Elaborada pelo autor.

Se considerarmos $\bar{x} = 1 - x$, podemos reescrever a função objetivo de (27) como

$$(MC_{\{0,1\}}) \quad \max \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}(x_i \bar{x}_j + \bar{x}_i x_j) \quad (28)$$

s.a $\bar{x}_j = 1 - x_j \quad \forall j = 1, \dots, n$
 $x \in \{0, 1\}^n$.

A formulação (28) será a base para a aplicação da t -linearização ao Max-Cut. Ver Seção 3.5.

3.1.3 Formulação binária $\{-1, 1\}$

Uma forma diferente de enxergar e formular o problema do Max-Cut se dá quando passamos a considerar o vetor binário $x \in \{-1, 1\}^n$, com $x_i = 1$ se $i \in S$ e $x_i = -1$ se $i \in \bar{S}$. A mudança de variáveis

$$x_i \leftarrow 2x_i - 1 \quad (29)$$

leva a uma expressão equivalente da função objetivo de (27) como mostra a Tabela 3.

Tabela 3: Peso do corte da (S, \bar{S}) representado pelo vetor binário $x \in \{-1, 1\}^n$

		Função Objetivo
x_i	x_j	$c_{ij} \left(\frac{1 - x_i x_j}{2} \right)$
-1	-1	0
-1	1	c_{ij}
1	-1	c_{ij}
1	1	0

Fonte: Elaborada pelo autor.

Assim podemos modelar o problema Max-Cut da seguinte forma:

$$\begin{aligned}
 (MC_{\{-1,1\}}) \quad & \max \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} \left(\frac{1 - x_i x_j}{2} \right) \\
 & \text{s.a. } x_i \in \{-1, 1\} \quad \forall i \in V.
 \end{aligned} \quad (30)$$

Note que as restrições em (30) podem ser substituídas por $x_i^2 = 1, \forall i \in V$.

3.2 Indexação nos Vértices

Nesta e na próxima seção, vamos derivar propriedades sobre as soluções ótimas do problema do corte máximo, ainda não estudadas na literatura. Tais propriedades permitem a fixação de variáveis e a derivação de desigualdades válidas para o problema. Nesta seção, expressamos tais resultados para a formulação (28), cujas variáveis são indexadas pelos vértices do grafo. Sua adaptação para a formulação (30) é direta, usando a transformação linear (29) entre as variáveis das duas formulações.

3.2.1 Condições de otimalidade

Consideramos a formulação para Max-Cut expressa em (28) em termos da variável $x \in \{0, 1\}^n$, onde x_i marca a partição que o vértice i pertence. A função objetivo é

$$Q(x) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}(x_i \bar{x}_j + \bar{x}_i x_j).$$

Note que

$$\begin{aligned} \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}(x_i \bar{x}_j + \bar{x}_i x_j) &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} x_i \bar{x}_j + \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} \bar{x}_i x_j \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} x_i \bar{x}_j + \sum_{j=1}^{n-1} \sum_{i=j+1}^n c_{ji} \bar{x}_j x_i \\ &= \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n c_{ij} x_i \bar{x}_j \quad (\text{usando } c_{ji} = c_{ij}) \end{aligned} \quad (31)$$

Dados $x \in \{0, 1\}^n$ e $k \in \{1, 2, \dots, n\}$, seja $x^{\bar{k}} \in \{0, 1\}^n$ tal que

$$x_i^{\bar{k}} = \begin{cases} x_i, & i \neq k, \\ 1 - x_i, & i = k. \end{cases}$$

Defina

$$C_k = \sum_{j \neq k} c_{kj} \quad \text{e} \quad C_k(x) = \sum_{j \neq k} c_{kj} x_j.$$

Determinamos a seguir a relação entre os valores de $Q(x)$ e $Q(x^{\bar{k}})$.

Lema 3.1. Para todo $x \in \{0, 1\}^n$ e todo $k \in \{1, 2, \dots, n\}$, $Q(x^{\bar{k}}) - Q(x) = \sum_{j \neq k} c_{kj}(1 - 2x_k)(1 - 2x_j)$.

Demonstração. Usando (31), temos que

$$\begin{aligned} Q(x^{\bar{k}}) - Q(x) &= \sum_{j \neq k} c_{kj} [x_k^{\bar{k}}(1 - x_j^{\bar{k}}) - x_k(1 - x_j)] + \sum_{i \neq k} c_{ik} [x_i^{\bar{k}}(1 - x_k^{\bar{k}}) - x_i(1 - x_k)] \\ &= \sum_{j \neq k} c_{kj} [(1 - x_k)(1 - x_j) - x_k(1 - x_j)] + \sum_{i \neq k} c_{ik} [x_i x_k - x_i(1 - x_k)] \\ &= \sum_{j \neq k} c_{kj} (1 - 2x_k)(1 - x_j) + \sum_{i \neq k} c_{ki} x_i (-1 + 2x_k) \\ &= \sum_{j \neq k} c_{kj} (1 - 2x_k)(1 - 2x_j). \end{aligned}$$

□

O lema acima leva às seguintes propriedades de uma solução ótima.

Corolário 3.1. *Seja x^* solução ótima para (28) e $k \in \{1, 2, \dots, n\}$. Se $x_k^* = 1$ então $C_k(x^*) \leq C_k/2$. Por outro lado, se $C_k(x^*) < C_k/2$ então $x_k^* = 1$.*

Demonstração. Defina $\bar{x} = (x^*)^{\bar{k}}$. Temos que $Q(\bar{x}) \leq Q(x^*)$. Primeiro, considere que $x_k^* = 1$. Pelo Lema 3.1, segue que

$$\begin{aligned} 0 \geq Q(\bar{x}) - Q(x^*) &= \sum_{j \neq k} -c_{kj}(1 - 2x_j^*) \\ &= -\sum_{j \neq k} c_{kj} + 2 \sum_{j \neq k} c_{kj}x_j^* \\ &= -C_k + 2C_k(x^*), \end{aligned}$$

mostrando a primeira implicação. Para mostrar a contrapositiva da segunda, considere que $x_k^* = 0$. Novamente pelo Lema 3.1, temos

$$Q(\bar{x}) - Q(x^*) = \sum_{j \neq k} c_{kj}(1 - 2x_j) = C_k - 2C_k(x^*) \leq 0.$$

□

O corolário acima pode ser reescrito alternativamente como:

Corolário 3.2. *Seja x^* solução ótima para (28) e $k \in \{1, 2, \dots, n\}$. Se $x_k^* = 0$ então $C_k(x^*) \geq C_k/2$. Por outro lado, se $C_k(x^*) > C_k/2$ então $x_k^* = 0$.*

Podemos estender as propriedades estabelecidas pelos corolários 3.1 e 3.2 para soluções ótimas de subproblemas de Max-Cut, obtidos com a fixação de alguns vértices em S ou \bar{S} . Para isso, apresentamos o conceito de solução parcial.

Definição 3.1. *Uma solução parcial para a formulação (28) é um vetor com entradas fixadas em 0 ou 1 ou não fixadas, ou seja, $x_j \in \{0, 1\}$ apenas para $j \in J' \subseteq \{1, \dots, n\}$.*

Definindo $J = \{1, \dots, n\}$, uma solução parcial x pode ser descrita pelos conjuntos $J_0(x) = \{j \in J : x_j = 0\}$, $J_1(x) = \{j \in J : x_j = 1\}$ e $J_2(x) = J \setminus (J_0(x) \cup J_1(x))$.

Definição 3.2. *Uma solução completa é uma solução parcial x onde $J_2(x) = \emptyset$.*

Definição 3.3. *Uma solução descendente de uma solução parcial x é uma solução parcial y tal que $J_0(y) \supseteq J_0(x)$, $J_1(y) \supseteq J_1(x)$ e $J_2(y) \neq J_2(x)$.*

A extensão dos corolários 3.1 e 3.2 para soluções parciais é apresentado a seguir.

Corolário 3.3. *Sejam \hat{x} uma solução parcial, x^* solução descendente de \hat{x} de maior valor e $k \in J_2(\hat{x})$. Se $x_k^* = 1$ então $2 \sum_{j \in J_2(\hat{x})} c_{kj}x_j^* \leq \sum_{j \in J_0(\hat{x}) \cup J_2(\hat{x}) \setminus \{k\}} c_{kj} - \sum_{j \in J_1(\hat{x})} c_{kj}$. Por outro lado, se $2 \sum_{j \in J_2(\hat{x})} c_{kj}x_j^* < \sum_{j \in J_0(\hat{x}) \cup J_2(\hat{x}) \setminus \{k\}} c_{kj} - \sum_{j \in J_1(\hat{x})} c_{kj}$ então $x_k^* = 1$.*

Corolário 3.4. *Sejam \hat{x} uma solução parcial, x^* solução descendente de \hat{x} de maior valor e $k \in J_2(\hat{x})$. Se $x_k^* = 0$ então $2 \sum_{j \in J_2(\hat{x})} c_{kj}x_j^* \geq \sum_{j \in J_0(\hat{x}) \cup J_2(\hat{x}) \setminus \{k\}} c_{kj} - \sum_{j \in J_1(\hat{x})} c_{kj}$. Por outro*

lado, se $2 \sum_{j \in J_2(\hat{x})} c_{kj} x_j^* > \sum_{j \in J_0(\hat{x}) \cup J_2(\hat{x}) \setminus \{k\}} c_{kj} - \sum_{j \in J_1(\hat{x})} c_{kj}$ então $x_k^* = 0$.

As demonstrações dos corolários 3.3 e 3.4 serão omitidas, tendo em vista que seguem a mesma ideia de prova dos corolários 3.1 e 3.2, bastando observar que $(x)^{\bar{k}}$ é descendente de \hat{x} pois $k \in J_2(\hat{x})$.

3.2.2 Fixação de variáveis

As propriedades estabelecidas na Subseção 3.2.1 nos permitem estabelecer critérios para fixação de variáveis em um processo de busca de solução ótima do Max-Cut.

Teorema 3.2. *Seja \hat{x} uma solução parcial e $k \in J_2(\hat{x})$. Dentre as soluções descendentes de \hat{x} de maior valor, existe uma, x , tal que $x_k = 1$ se*

$$\sum_{j \in J_1(\hat{x}) \cup J_0(\hat{x})} (1 - 2\hat{x}_j) c_{kj} - \sum_{j \in J_2(\hat{x}) \setminus \{k\}} |c_{kj}| \geq 0. \quad (32)$$

Demonstração. Tome solução x descendente de \hat{x} tal que $x_k = 0$. Note que $x^{\bar{k}}$ também é descendente de \hat{x} , pois $k \in J_2(\hat{x})$. Pelo Lema 3.1, temos

$$\begin{aligned} Q(x^{\bar{k}}) - Q(x) &= \sum_{j \neq k} c_{kj} (1 - 2x_k)(1 - 2x_j) = \sum_{j \neq k} c_{kj} (1 - 2x_j) \\ &= \sum_{j \in J_0(\hat{x}) \cup J_1(\hat{x})} (1 - 2\hat{x}_j) c_{kj} + \sum_{j \in J_2(\hat{x}) \setminus \{k\}} c_{kj} (1 - 2x_j) \\ &= \sum_{j \in J_0(\hat{x}) \cup J_1(\hat{x})} (1 - 2\hat{x}_j) c_{kj} + \sum_{j \in J_2(\hat{x}) \setminus \{k\}} c_{kj} - 2 \sum_{j \in J_2(\hat{x}) \setminus \{k\}} c_{kj} x_j \\ &\geq \sum_{j \in J_0(\hat{x}) \cup J_1(\hat{x})} (1 - 2\hat{x}_j) c_{kj} + \sum_{j \in J_2(\hat{x}) \setminus \{k\}} c_{kj} - 2 \sum_{j \in J_2(\hat{x}) \setminus \{k\}} \max\{0, c_{kj}\} \\ &= \sum_{j \in J_0(\hat{x}) \cup J_1(\hat{x})} (1 - 2\hat{x}_j) c_{kj} - \sum_{j \in J_2(\hat{x}) \setminus \{k\}} |c_{kj}| \geq 0. \end{aligned}$$

Logo, $Q(x^{\bar{k}}) \geq Q(x)$ levando ao resultado desejado. \square

A condição (32) pode ser reescrita como:

$$\sum_{j \in J_0(\hat{x})} c_{kj} + \sum_{j \in J_2(\hat{x}) \setminus \{k\}} c_{kj}^- \geq \sum_{j \in J_1(\hat{x})} c_{kj} + \sum_{j \in J_2(\hat{x}) \setminus \{k\}} c_{kj}^+.$$

O lado esquerdo da expressão acima somado a $\sum_{(i,j) \in J_0(\hat{x}) \times J_1(\hat{x})} c_{ij}$ é um limite inferior para

as soluções descendentes de \hat{x} onde $x_k = 1$. Por outro lado, o termo da direita somado a

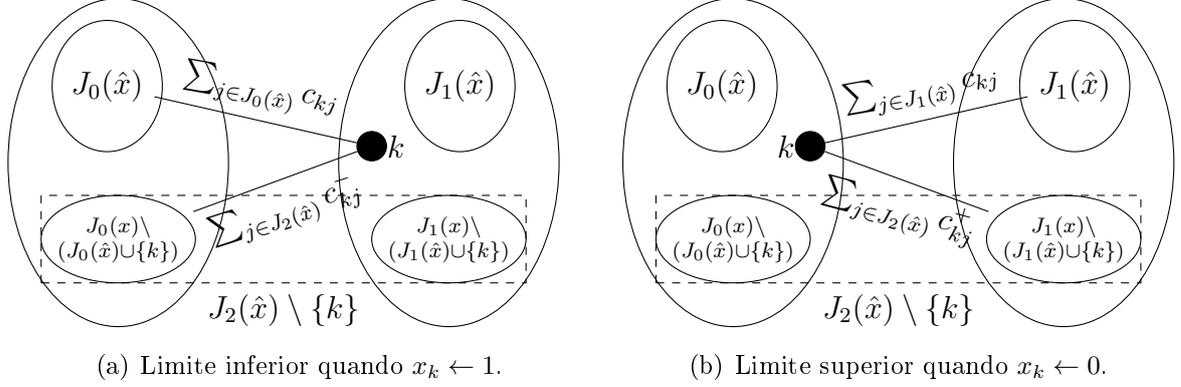
$\sum_{(i,j) \in J_0(\hat{x}) \times J_1(\hat{x})} c_{ij}$ é um limite superior para as soluções descendentes de \hat{x} onde $x_k = 0$. Ver

Figura 8 para uma ilustração. Dessa forma, a condição (32) é satisfeita quando esse limite inferior é maior que ou igual a esse limite superior, levando ao resultado do Teorema 3.2.

Similarmente, podemos definir critérios para fixação de variáveis em 0, como

a seguir.

Figura 8: Fixação $x_k = 1$ em uma solução descendente de \hat{x} : limite inferior (a) maior que limite superior (b).



Fonte: Elaborado pelo autor.

Teorema 3.3. *Seja \hat{x} uma solução parcial e $k \in J_2(\hat{x})$. Dentre as soluções descendentes de \hat{x} de maior valor, existe uma, x , tal que $x_k = 0$ se*

$$\sum_{j \in J_1(\hat{x}) \cup J_0(\hat{x})} (1 - 2\hat{x}_j) c_{kj} + \sum_{j \in J_2(\hat{x}) \setminus \{k\}} |c_{kj}| \leq 0. \quad (33)$$

Demonstração. Tome solução x descendente de \hat{x} tal que $x_k = 1$. Note que $x^{\bar{k}}$ também é descendente de \hat{x} , pois $k \in J_2(\hat{x})$. Usando o Lema 3.1, obtemos

$$\begin{aligned} Q(x^{\bar{k}}) - Q(x) &= \sum_{j \neq k} c_{kj} (1 - 2x_k)(1 - 2x_j) = \sum_{j \neq k} -c_{kj} (1 - 2x_j) \\ &= \sum_{j \in J_0(\hat{x}) \cup J_1(\hat{x})} (-1 + 2\hat{x}_j) c_{kj} + \sum_{j \in J_2(\hat{x}) \setminus \{k\}} -c_{kj} (1 - 2x_j) \\ &= \sum_{j \in J_0(\hat{x}) \cup J_1(\hat{x})} (-1 + 2\hat{x}_j) c_{kj} + \sum_{j \in J_2(\hat{x}) \setminus \{k\}} -c_{kj} + 2 \sum_{j \in J_2(\hat{x}) \setminus \{k\}} c_{kj} x_j \\ &\geq \sum_{j \in J_0(\hat{x}) \cup J_1(\hat{x})} (-1 + 2\hat{x}_j) c_{kj} + \sum_{j \in J_2(\hat{x}) \setminus \{k\}} -c_{kj} + 2 \sum_{j \in J_2(\hat{x}) \setminus \{k\}} \min\{0, c_{kj}\} \\ &= \sum_{j \in J_0(\hat{x}) \cup J_1(\hat{x})} (-1 + 2\hat{x}_j) c_{kj} - \sum_{j \in J_2(\hat{x}) \setminus \{k\}} |c_{kj}| \geq 0. \end{aligned}$$

Concluimos que $Q(x^{\bar{k}}) \geq Q(x)$, mostrando o resultado desejado. \square

Os teoremas acima mostram que podemos fixar x_k (em 1 ou 0) na busca por soluções descendentes de \hat{x} , caso uma das condições (32) ou (33) seja satisfeita. Essas duas condições podem ser expressas conjuntamente como

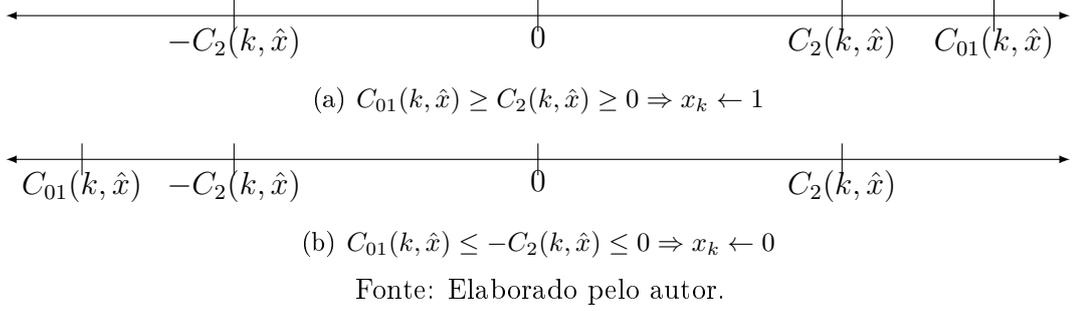
$$|C_{01}(k, \hat{x})| \geq C_2(k, \hat{x}), \quad (34)$$

onde

$$C_{01}(k, \hat{x}) = \sum_{j \in J_1(\hat{x}) \cup J_0(\hat{x})} (1 - 2\hat{x}_j)c_{kj} \quad \text{e} \quad C_2(k, \hat{x}) = \sum_{j \in J_2(\hat{x}) \setminus \{k\}} |c_{kj}|. \quad (35)$$

Assim, sendo satisfeita a condição (34), podemos fixar $x_k = 1$ (se $C_{01}(k, \hat{x}) \geq 0$) ou $x_k = 0$ (se $C_{01}(k, \hat{x}) \leq 0$). Ver Figura 9.

Figura 9: Fixação de variáveis.



3.2.3 Desigualdades válidas

Além de levar a fixação de variáveis, sob certas condições, as propriedades apresentadas na Subseção 3.2.1 permitem derivar desigualdades válidas para a formulação (28).

Note que as condições estabelecidas nos corolários 3.1 e 3.2 podem ser traduzidas nas seguintes desigualdades não lineares, válidas para as soluções ótimas de (28):

$$x_k(C_k/2 - C_k(x)) \geq 0 \quad \forall k \in \{1, \dots, n\}, \quad (36)$$

$$(x_k - 1)(C_k/2 - C_k(x)) \geq 0 \quad \forall k \in \{1, \dots, n\}. \quad (37)$$

Alternativamente, elas podem ser substituídas pelas desigualdades lineares

$$C_k(x) - M_k^1(1 - x_k) \leq \frac{C_k}{2}x_k \quad \forall k \in \{1, \dots, n\}, \quad (38)$$

$$C_k(x) + M_k^0x_k \geq \frac{C_k}{2}(1 - x_k) \quad \forall k \in \{1, \dots, n\}, \quad (39)$$

onde M_k^1 e M_k^0 são valores suficientemente grandes para tornar (38) e (39) redundantes quando $x_k = 0$ e $x_k = 1$, respectivamente.

Um valor para M_k^1 deve satisfazer $M_k^1 \geq \sum_{j \neq k} c_{kj}x_j$, para todo x ótimo de (28).

Então podemos tomar

$$M_k^1 \geq \sum_{j \neq k} \max\{0, c_{kj}\}. \quad (40)$$

Similarmente, $M_k^0 \geq -\sum_{j \neq k} c_{kj}x_j$ para todo x ótimo, de modo que podemos escolher

$$M_k^0 \geq -\sum_{j \neq k} \min\{0, c_{kj}\}. \quad (41)$$

Os valores de M_k^1 e M_k^0 podem ser reduzidos, a partir dos valores ótimos dos seguintes problemas:

$$\begin{aligned} (P_k^1) \quad & \max C_k(x) \\ & \text{s.a.} \quad M_j^1(1-x_j) \geq C_j(x) - \frac{C_j}{2}x_j \quad \forall j \neq k, \\ & \quad M_j^0x_j \geq -C_j(x) + \frac{C_j}{2}(1-x_j) \quad \forall j \neq k, \\ & \quad x \in \{0, 1\}^n, \quad x_k = 0. \end{aligned} \quad (42)$$

$$\begin{aligned} (P_k^0) \quad & \max -C_k(x) \\ & \text{s.a.} \quad M_j^1(1-x_j) \geq C_j(x) - \frac{C_j}{2}x_j \quad \forall j \neq k, \\ & \quad M_j^0x_j \geq -C_j(x) + \frac{C_j}{2}(1-x_j) \quad \forall j \neq k, \\ & \quad x \in \{0, 1\}^n, \quad x_k = 1. \end{aligned} \quad (43)$$

Mais precisamente, se LS_k^1 e LS_k^0 são limites superiores para P_k^1 e P_k^0 , respectivamente, podemos fazer $M_k^1 = \min\{M_k^1, LS_k^1\}$ e $M_k^0 = \min\{M_k^0, LS_k^0\}$. Na verdade, podemos definir os problemas P_k^1 e P_k^0 iterativamente à medida que melhores valores para M_i^1 e M_i^0 , $i \neq k$, forem sendo obtidos, de modo a potencialmente gerarem melhores valores para M_k^1 e M_k^0 .

Também podemos fortalecer as restrições (38) e (39) para obter desigualdades válidas apenas para a melhor solução descendente de uma solução parcial \bar{x} . É suficiente reduzir o valor de M_k^1 e M_k^0 . Inicialmente, podemos tomar

$$\begin{aligned} M_k^1 &\geq \sum_{j \in J_1(\bar{x})} c_{kj} + \sum_{j \in J_2(\bar{x}) \setminus \{k\}} \max\{0, c_{kj}\}, \quad \forall k \in J_2(\bar{x}), \\ M_k^0 &\geq -\sum_{j \in J_1(\bar{x})} c_{kj} - \sum_{j \in J_2(\bar{x}) \setminus \{k\}} \min\{0, c_{kj}\}, \quad \forall k \in J_2(\bar{x}). \end{aligned}$$

Iterativamente, podemos reduzir o valor de M_k^1 e M_k^0 , acrescentando as restrições $x_j = 0 \forall j \in J_0(\bar{x})$ e $x_j = 1 \forall j \in J_1(\bar{x})$ aos problemas (P_k^1) e (P_k^0) .

3.2.4 Cortes lógicos

Podemos estender o resultado do Lema 3.1, avaliando a mudança no valor da função quando trocamos o valor de não apenas uma mas de um subconjunto de variáveis.

Sejam $x \in \{0, 1\}^n$, $K \subseteq \{1, \dots, n\}$ e $\bar{K} = \{1, \dots, n\} \setminus K$. Defina $K_0(x) = \{k \in K : x_k = 0\}$, $K_1(x) = \{k \in K : x_k = 1\}$. Denote por $x^K \in \{0, 1\}^n$ o vetor tal que

$$x_i^K = \begin{cases} x_i, & i \in K, \\ 1 - x_i, & i \in \bar{K}. \end{cases} \quad (44)$$

Lema 3.2. Para todo $x \in \{0, 1\}^n$ e todo $K \subseteq \{1, \dots, n\}$, $\delta_k(x) := Q(x^K) - Q(x) = \sum_{i \notin K} \sum_{j \in K} c_{ij}(1 - 2x_i)(1 - 2x_j)$.

Demonstração. Usando (31), temos que

$$\begin{aligned}
\delta(x) &= \sum_{i \notin K} \sum_{\substack{j \in K \\ j \neq i}} c_{ij} [(1 - x_i)(1 - x_j) - x_i(1 - x_j)] + \sum_{i \in K} \sum_{\substack{j \notin K \\ j \neq i}} c_{ij} [(x_i x_j - x_i(1 - x_j))] \\
&+ \sum_{i \notin K} \sum_{\substack{j \notin K \\ j \neq i}} c_{ij} [(1 - x_i)x_j - x_i(1 - x_j)] \\
&= \sum_{i \notin K} \sum_{j \in K} c_{ij}(1 - 2x_i)(1 - x_j) + \sum_{i \in K} \sum_{j \notin K} c_{ij} x_i(-1 + 2x_j) + \underbrace{\sum_{i \notin K} \sum_{j \notin K} c_{ij}(x_j - x_i)}_0 \\
&= \sum_{i \notin K} \sum_{j \in K} c_{ij}(1 - 2x_i)(1 - x_j) + \sum_{i \notin K} \sum_{j \in K} c_{ij} x_j(-1 + 2x_i) \\
&= \sum_{i \notin K} \sum_{j \in K} c_{ij}(1 - 2x_i)(1 - 2x_j)
\end{aligned}$$

□

Dada uma solução corrente \hat{x} , podemos usar o lema acima para cortar soluções com valores menores que ou iguais a $Q(\hat{x})$.

Proposição 3.1. Sejam $\hat{x} \in \{0, 1\}^n$ e $K \subseteq \{1, \dots, n\}$. Defina $A = [\overline{K}_0(\hat{x}) \times K_0(\hat{x})] \cup [\overline{K}_1(\hat{x}) \times K_1(\hat{x})]$ e $B = [\overline{K}_0(\hat{x}) \times K_1(\hat{x})] \cup [\overline{K}_1(\hat{x}) \times K_0(\hat{x})]$. Se $\sum_{(i,j) \in A} c_{ij} - \sum_{(i,j) \in B} c_{ij} \leq 0$,

então toda solução x tal que $Q(x) > Q(\hat{x})$ deve satisfazer

$$\sum_{j \in K_1(\hat{x}) \cup \overline{K}_0(\hat{x})} (1 - x_j) + \sum_{j \in K_0(\hat{x}) \cup \overline{K}_1(\hat{x})} x_j \geq 1. \quad (45)$$

Demonstração. Suponha que $\sum_{(i,j) \in A} c_{ij} - \sum_{(i,j) \in B} c_{ij} \leq 0$. Seja $x \in \{0, 1\}^n$ que não satisfaz

(45). Então, $x_j = 1$, para todo $j \in K_1(\hat{x}) \cup \overline{K}_0(\hat{x})$, e $x_j = 0$, para todo $j \in K_0(\hat{x}) \cup \overline{K}_1(\hat{x})$.

Em outros termos,

$$\begin{aligned}
x_j &= \hat{x}_j \quad \forall j \in K, \\
x_j &= 1 - \hat{x}_j \quad \forall j \in \overline{K},
\end{aligned}$$

ou seja, $x = \hat{x}^K$. Queremos mostrar que $Q(x) \leq Q(\hat{x})$. Pelo Lema 3.2

$$\begin{aligned}
Q(\hat{x}^K) - Q(\hat{x}) &= \sum_{(i,j) \in \bar{K} \times K} c_{ij}(1 - 2\hat{x}_i)(1 - 2\hat{x}_j) \\
&= \sum_{(i,j) \in \bar{K}_0(\hat{x}) \times K_0(\hat{x})} c_{ij} + \sum_{(i,j) \in \bar{K}_1(\hat{x}) \cup K_1(\hat{x})} c_{ij} + \sum_{(i,j) \in \bar{K}_0(\hat{x}) \times K_1(\hat{x})} -c_{ij} + \sum_{(i,j) \in \bar{K}_1(\hat{x}) \times K_0(\hat{x})} -c_{ij} \\
&= \sum_{(i,j) \in A} c_{ij} - \sum_{(i,j) \in B} c_{ij} \leq 0
\end{aligned}$$

Logo $Q(x) \leq Q(\hat{x})$. □

3.3 Indexação nas Arestas

Nesta seção, adaptamos o conteúdo da Seção 3.2 para a formulação (26) do Max-Cut com variáveis indexadas pelas arestas do grafo.

3.3.1 Condições de otimalidade

Consideramos aqui a formulação (26), definida em termos das variáveis $z \in \{0, 1\}^m$, que marcam se uma aresta pertence ($z_{ij} = 1$) ou não ($z_{ij} = 0$) ao corte. Pelas restrições triangulares

$$z_{ij} + z_{ik} + z_{jk} \leq 2, \quad (46)$$

$$z_{ij} - z_{ik} - z_{jk} \leq 0, \quad (47)$$

$$-z_{ij} + z_{ik} - z_{jk} \leq 0, \quad (48)$$

$$-z_{ij} - z_{ik} + z_{jk} \leq 0, \quad (49)$$

temos que

$$1. \ z_{ij} = 0 \Rightarrow z_{ik} = z_{jk} \quad \forall k \neq i, j \text{ (por (48) e (49))}$$

$$2. \ z_{ij} = 1 \Rightarrow z_{ik} + z_{jk} = 1 \quad \forall k \neq i, j \text{ (por (46))}$$

Pelas definições das variáveis z e x (da formulação (27)), temos também a seguinte relação:

$$z_{ij} = 0 \Leftrightarrow x_i = x_j \quad \text{e} \quad z_{ij} = 1 \Leftrightarrow x_i \neq x_j.$$

Na verdade, se um vetor $z \in \{0, 1\}^m$ descreve um corte (S, \bar{S}) , então ele corresponde a dois vetores $x \in \{0, 1\}^n$ e $x' \in \{0, 1\}^n$ que descrevem esse corte, sendo $x'_j = 1 - x_j \quad \forall j$ e $x_j = 1 \Leftrightarrow j \in S$. Dizemos que x e x' são soluções viáveis para (27) correspondentes à solução z viável para (26).

Podemos escrever uma versão do Lema 3.1 para a formulação (26).

Lema 3.3. *Sejam z e x soluções correspondentes e $k \in \{1, \dots, n\}$. Então, $Q(x^{\bar{k}}) - Q(x) = \sum_{j \neq k} c_{kj}(1 - 2z_{ij})$.*

Demonstração. Pelo Lema 3.1, temos $Q(x^{\bar{k}}) - Q(x) = \sum_{j \neq k} c_{kj}(1 - 2x_k)(1 - 2x_j)$. Note que

$$\begin{aligned} (1 - 2x_k)(1 - 2x_j) &= (1 - 2x_k - 2x_j + 4x_kx_j) \\ &= \begin{cases} 1, & \text{se } x_k = x_j \\ -1, & \text{se } x_k \neq x_j \end{cases} \\ &= (1 - 2z_{kj}) \end{aligned}$$

Portanto, em termos da variável z , temos que $Q(x^{\bar{k}}) - Q(x) = \sum_{j \neq k} c_{kj}(1 - 2z_{ij})$. \square

Corolário 3.5. *Seja z^* solução ótima para (26) e $k, e \in \{1, \dots, n\}, k \neq e$. Se $z_{ke}^* = 0$ então $2 \sum_{j \neq k, e} c_{kj}z_{kj}^* = 2 \sum_{j \neq k, e} c_{kj}z_{ej}^* \geq \sum_{j \neq k} c_{kj}$. Por outro lado, se $2 \sum_{j \neq k, e} c_{kj}z_{ej}^* > \sum_{j \neq k} c_{kj}$ então $z_{ke}^* = 0$.*

Demonstração. Sejam x^* uma solução correspondente a z^* e $\bar{x} = (x^*)^{\bar{k}}$. Então $Q(\bar{x}) \leq Q(x^*)$. Logo, pelo Lema 3.3

$$0 \geq Q(\bar{x}) - Q(x^*) = \sum_{j \neq k} c_{kj}(1 - 2z_{ij}^*) = \sum_{j \neq k} c_{kj} - 2 \sum_{j \neq k, e} c_{kj}z_{kj}^* - 2c_{ke}z_{ke}^*. \quad (50)$$

Suponha que $z_{ke}^* = 0$. Então $z_{kj}^* = z_{ej}^* \forall j \neq k, e$ e, assim, por (50), a primeira implicação segue. Suponha agora que $z_{ke}^* = 1$, o que implica $z_{kj}^* = 1 - z_{ej}^*, \forall j \neq k, e$. Por (50), temos

$$\begin{aligned} 0 \geq Q(\bar{x}) - Q(x^*) &= \sum_{j \neq k} c_{kj} - 2 \sum_{j \neq k, e} c_{kj}z_{kj}^* - 2c_{ke} \\ &= \sum_{j \neq k} c_{kj} - 2 \sum_{j \neq k, e} c_{kj} + 2 \sum_{j \neq k, e} c_{kj}z_{ej}^* - 2c_{ke} \\ &= - \sum_{j \neq k} c_{kj} + 2 \sum_{j \neq k, e} c_{kj}z_{ej}^* \end{aligned}$$

Assim obtemos a segunda implicação. \square

Corolário 3.6. *Seja z^* solução ótima para (26) e $k, e \in \{1, \dots, n\}, k \neq e$. Se $z_{ke}^* = 1$ então $\sum_{j \neq k} c_{kj} \geq 2 \sum_{j \neq k, e} c_{kj}z_{ej}^*$. Por outro lado, se $\sum_{j \neq k} c_{kj} > 2 \sum_{j \neq k, e} c_{kj}z_{ej}^*$ então $z_{ke}^* = 1$.*

Similarmente aos corolários 3.3 e 3.4, podemos obter versões dos corolários 3.5 e 3.6 referentes a soluções parciais de (26), definidas de forma análoga.

3.3.2 Desigualdades válidas

Os corolários 3.5 e 3.6 fornecem as seguintes restrições válidas para soluções ótimas de (26), para todos k, e com $k \neq e$:

$$C_{ke}(z) + M_{ke}^0 z_{ke} \geq \frac{C_k}{2}(1 - z_{ke}), \quad (51)$$

$$C_{ek}(z) + M_{ek}^0 z_{ke} \geq \frac{C_e}{2}(1 - z_{ke}), \quad (52)$$

$$C_{ke}(z) + M_{ke}^1(1 - z_{ke}) \leq \frac{C_k}{2} z_{ke}, \quad (53)$$

$$C_{ek}(z) + M_{ek}^1(1 - z_{ke}) \leq \frac{C_e}{2} z_{ke}, \quad (54)$$

onde $C_k = \sum_{j \neq k} c_{kj}$, $C_{ke}(z) = \sum_{j \neq k, e} c_{kj} z_{ej}$, $C_{ek}(z) = \sum_{j \neq k, e} c_{ej} z_{kj}$ e M_{ke}^0 , M_{ek}^0 , M_{ke}^1 , M_{ek}^1 são valores grandes o suficientes para tornar as restrições redundantes nos casos adequados.

Os valores de M^0 e M^1 podem ser reduzidos, considerando os seguintes problemas, para todos k, e com $k \neq e$:

$$\begin{aligned} (P_{ke}^0) \quad & \max \quad -C_{ke}(z) \\ \text{s.a.} \quad & C_{ij}(z) + M_{ij}^0 z_{ij} \geq \frac{C_i}{2}(1 - z_{ij}) \quad \forall (i, j) \neq (k, e), \\ & C_{ij}(z) + M_{ij}^1(1 - z_{ij}) \leq \frac{C_i}{2} z_{ij} \quad \forall (i, j) \neq (k, e), \\ & z_{ij} \in \{0, 1\}, \quad z_{ke} = 1, \quad \forall (i, j) \neq (k, e). \end{aligned} \quad (55)$$

$$\begin{aligned} (P_{ke}^1) \quad & \max \quad C_{ke}(z) \\ \text{s.a.} \quad & C_{ij}(z) + M_{ij}^0 z_{ij} \geq \frac{C_i}{2}(1 - z_{ij}) \quad \forall (i, j) \neq (k, e), \\ & C_{ij}(z) + M_{ij}^1(1 - z_{ij}) \leq \frac{C_i}{2} z_{ij} \quad \forall (i, j) \neq (k, e), \\ & z_{ij} \in \{0, 1\}, \quad z_{ke} = 0 \quad \forall (i, j) \neq (k, e). \end{aligned} \quad (56)$$

Se LS_{ke}^0 e LS_{ke}^1 são limites superiores para P_{ke}^0 e P_{ke}^1 , respectivamente, podemos fazer $M_{ke}^0 = \min\{LS_{ke}^0, M_{ke}^0\}$ e $M_{ke}^1 = \min\{LS_{ke}^1, M_{ke}^1\}$. Na verdade, podemos definir os problemas P_{ke}^0 e P_{ke}^1 iterativamente, à medida que melhores valores para M_{ij}^0 e M_{ij}^1 , $(i, j) \neq (k, e)$, forem sendo obtidos, de modo a fornecerem melhores valores para M_{ke}^0 e M_{ke}^1 .

3.4 Aplicação das Linearizações Clássica e de Glover

Vamos aplicar as linearizações Clássica e Glover à formulação (28). Relembre que a função

objetivo de (28) é $\sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}(x_i \bar{x}_j + \bar{x}_i x_j)$. Por conveniência, vamos reescrevê-la como:

$$\begin{aligned} \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}(x_i \bar{x}_j + \bar{x}_i x_j) &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n -2c_{ij}x_i x_j + \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}x_i + \sum_{j=1}^{n-1} \sum_{i=j+1}^n c_{ji}x_i \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n -2c_{ij}x_i x_j + \sum_{i=1}^n \sum_{j=1}^n c_{ij}x_i \quad (\text{usando } c_{ii} = 0). \end{aligned} \quad (57)$$

Aplicando as linearizações vistas nas subseções 2.1 e 2.2, obtemos os seguintes

modelos para o Max-Cut:

$$\begin{aligned}
(MC)_{lc} \quad & \max \sum_{i=1}^{n-1} \sum_{j=i+1}^n -2c_{ij}y_{ij} + \sum_{i=1}^n \sum_{j=1}^n c_{ij}x_i \\
s.a \quad & y_{ij} - x_i \leq 0 \quad \forall 1 \leq i < j \leq n, \\
& y_{ij} - x_j \leq 0 \quad \forall 1 \leq i < j \leq n, \\
& -y_{ij} + x_i + x_j \leq 1 \quad \forall 1 \leq i < j \leq n, \\
& y_{ij} \in \{0, 1\} \quad \forall 1 \leq i < j \leq n, \\
& x \in \{0, 1\}^n.
\end{aligned} \tag{58}$$

$$\begin{aligned}
(MC)_{lg} \quad & \max \sum_{i=1}^{n-1} u_i + \sum_{i=1}^n \sum_{j=1}^n c_{ij}x_i \\
s.a \quad & u_i + \sum_{j=i+1}^n 2c_{ij}^- x_i \leq 0 \quad 1 \leq i \leq n-1, \\
& u_i + \sum_{j=i+1}^n 2c_{ij} x_j \leq \sum_{j=i+1}^n 2c_{ij}^+ (1-x_i) \quad 1 \leq i \leq n-1, \\
& x \in \{0, 1\}^n.
\end{aligned} \tag{59}$$

As relaxações lineares dessas formulações fornecem limites superiores para o Max-Cut. Mostramos a seguir que esses limites são iguais ao limite trivial, dado pela soma dos custos positivos.

Teorema 3.4. *Os valores ótimos das relaxações de (58) e (59) são $\sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}^+$.*

Demonstração. Sejam V_{lc} e V_{lg} os valores ótimos das relaxações lineares de (58) e (59), respectivamente. Pelo Teorema 2.1, temos que $V_{lg} \geq V_{lc}$. Agora observe que (x^*, y^*) tal que $x_i^* = 0.5$, $y_{ij}^* = 0$ (se $c_{ij} \geq 0$) e $y_{ij}^* = 0.5$ (se $c_{ij} < 0$), para todo $i, j \in \{1, \dots, n\}$, $j > i$,

é solução viável para (58). Note ainda que $2c_{ij}y_{ij} = c_{ij}^-$. Então, $V_{lc} \geq \sum_{i=1}^{n-1} \sum_{j=i+1}^n -2c_{ij}y_{ij}^* +$

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij}x_i^* = \sum_{i=1}^{n-1} \sum_{j=i+1}^n -c_{ij}^- + \sum_{i=1}^n \sum_{j=i+1}^n c_{ij} = \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}^+.$$

Para limitar superiormente V_{lg} , vamos considerar o dual da relaxação linear de (59). Definindo uma variável dual associada à cada restrição da relaxação, inclusive a $x_i \leq 1$, para todo $i \in \{1, \dots, n\}$, obtemos o seguinte problema dual:

$$\begin{aligned}
(DMC)_{lg} \quad & \max \sum_{i=1}^{n-1} \left(\sum_{j=i+1}^n 2c_{ij}^+ \right) \alpha_i + \sum_{i=1}^n \gamma_i \\
s.a \quad & \alpha_i + \beta_i = 1 \quad \forall i \in \{1, \dots, n-1\}, \\
& \sum_{j=i+1}^n 2c_{ij}^- \alpha_i + \sum_{j=i+1}^n 2c_{ij}^+ \beta_i + \sum_{j=1}^{i-1} 2c_{ji} \beta_j + \gamma_i \geq \sum_{j=1}^n c_{ij} \\
& \quad \forall i \in \{1, \dots, n-1\}, \\
& \alpha_i, \beta_i \geq 0 \quad \forall i \in \{1, \dots, n-1\}, \\
& \gamma_i \geq 0 \quad \forall i \in \{1, \dots, n\}.
\end{aligned}$$

Temos que $(\alpha^*, \beta^*, \gamma^*)$ tal que $\alpha_i^* = \beta_i^* = 0.5$ e $\gamma_i^* = 0$ é viável para $(DMC)_{lg}$.

Pelo teorema de dualidade fraca, $V_{lg} \leq \sum_{i=1}^{n-1} \left(\sum_{j=i+1}^n 2c_{ij}^+ \right) \alpha_i^* + \sum_{i=1}^n \gamma_i^* = \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}^+$. \square

3.5 Aplicação da t -Linearização ao Max-Cut

Nesta seção, vamos aplicar a t -linearização à formulação (28) do Max-Cut. Por (57), a função objetivo de (28) é

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} (x_i \bar{x}_j + \bar{x}_i x_j) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n -2c_{ij} x_i x_j + \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_i.$$

A expressão à direita tem a mesma forma de (13). Então, aplicando o desenvolvimento da Subseção 2.3.2, transferimos o termo quadrático da função objetivo para a restrição, usando uma variável t , como segue:

$$\begin{aligned}
\max \quad & t + \sum_{i=1}^n \sum_{j=1}^n c_{ij}^- x_i \\
t \leq \quad & \sum_{i=1}^n \sum_{j=1}^{i-1} -2c_{ij}^- x_i x_j + \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ij}^+ (x_i \bar{x}_j + \bar{x}_i x_j), \\
x \in \quad & \{0, 1\}^n, \quad \bar{x}_j = 1 - x_j \quad \forall j = 1, \dots, n.
\end{aligned}$$

Note que $(-d)^+ = -d^-$ e $(-d)^- = -d^+$, para todo $d \in \mathbb{R}$.

Em seguida, substituímos essa restrição quadrática por um conjunto de restrições lineares, dado pelo Teorema 2.5, obtendo assim a seguinte linearização para o Max-Cut:

$$\begin{aligned}
(MC)_\pi^t \quad \max \quad & t + \sum_{i=1}^n \sum_{j=1}^n c_{ij}^- x_i \\
s.a \quad & t \leq \sum_i \sum_{\substack{j < i \\ \pi(j) \in N_1}} c_{\pi(j)\pi(i)}^+ (\bar{x}_{\pi(i)} + \bar{x}_{\pi(j)}) + \sum_i \sum_{\substack{j < i \\ \pi(j) \in N_0}} c_{\pi(j)\pi(i)}^+ (x_{\pi(i)} + x_{\pi(j)}) + \\
& \sum_i \sum_{\substack{j < i \\ \pi(j) \in N_1}} -2c_{\pi(j)\pi(i)}^- x_{\pi(i)} + \sum_i \sum_{\substack{j < i \\ \pi(j) \in N_0}} -2c_{\pi(j)\pi(i)}^- x_{\pi(j)} \forall \pi \in S_n \forall (N_0, N_1) \in \mathcal{N}, \\
& (x, t) \in \mathbb{B}^n \times \mathbb{R}, \quad \bar{x}_j = 1 - x_j \quad \forall j = 1, \dots, n.
\end{aligned} \tag{60}$$

Mostramos que o valor da relaxação linear da formulação t -linearizada de Max-Cut coincide com aquele da relaxação clássica.

Teorema 3.5. *O valor ótimo da relaxação linear de (60) é igual a $\sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}^+$.*

Demonstração. Seja $x \in \mathbb{R}^n$ tal que $x_i = 0.5$, para todo $i \in \{1, \dots, n\}$. Pelo Teorema 2.6, podemos deduzir que (x, t) é viável para a relaxação de (60), onde t é o valor ótimo do problema de separação (15), ou seja, $t = \sum_i \sum_{j < i} c_{ji}^+ + \sum_i \sum_{j < i} -c_{ji}^-$. Logo, o valor da relaxação linear é pelo menos o valor dessa solução, que é dado por

$$t + \sum_{i=1}^n \sum_{j=1}^n c_{ij}^- / 2 = t + \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ij}^- = \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}^+.$$

Adicionalmente, esse é o valor da relaxação da linearização clássica, conforme o Teorema 3.4. Logo, o resultado segue pelo Teorema 2.7. \square

3.6 Aplicação da Suavização Hiperbólica ao Max-Cut

Nessa seção, iremos aplicar o desenvolvimento da suavização hiperbólica ao problema do corte máximo. Usaremos a formulação (27), definida a partir das variáveis $x \in \mathbb{B}^n$, junto com as desigualdades triangulares da formulação (26), ou seja,

$$\begin{aligned}
z_{ij} + z_{ik} + z_{jk} &\leq 2, & \forall \{i, j, k\} \subseteq V, & \quad i < j < k, \\
z_{ij} - z_{ik} - z_{jk} &\leq 0, & \forall \{i, j, k\} \subseteq V, & \quad i < j < k, \\
-z_{ij} + z_{ik} - z_{jk} &\leq 0, & \forall \{i, j, k\} \subseteq V, & \quad i < j < k, \\
-z_{ij} - z_{ik} + z_{jk} &\leq 0, & \forall \{i, j, k\} \subseteq V, & \quad i < j < k.
\end{aligned}$$

Lembre que z_{ij} é a variável que indica se a aresta formada pelos vértices $\{i, j\}$ pertence ao corte do grafo. Já na formulação (27), a variável $x_i \in \{0, 1\}$ nos diz à qual partição pertence o vértice i , de modo que a expressão $(x_i + x_j - 2x_i x_j)$ fornece valor correspondente

a z_{ij} . Mais ainda, como $x_i = x_i^2 \forall i$, temos que

$$x_i + x_j - 2x_i x_j = x_i^2 - 2x_i x_j + x_j^2 = (x_i - x_j)^2 = |x_i - x_j|.$$

A Tabela 4 mostra as equivalências entre z_{ij} , $x_i + x_j - 2x_i x_j$ e $|x_i - x_j|$.

Tabela 4: Equivalência entre $x_i + x_j - 2x_i x_j$, $|x_i - x_j|$ e z_{ij}

Vértices		Relação de Equivalência			Descrição
x_i	x_j	$x_i + x_j - 2x_i x_j$	$ x_i - x_j $	z_{ij}	aresta $\{i, j\}$
0	0	0	0	0	\notin corte
0	1	1	1	1	\in corte
1	0	1	1	1	\in corte
1	1	0	0	0	\notin corte

Fonte: Elaborada pelo autor.

Observe então que o seguinte modelo é uma relaxação para (27):

$$\begin{aligned}
(|MC|) \quad \min \quad & - \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} |x_i - x_j| \\
s.a \quad & 2(x_i + x_j + x_k) - 2(x_i x_j + x_i x_k + x_j x_k) - 2 \leq 0, \quad i < j < k, \\
& 2(-x_k - x_i x_j + x_i x_k + x_j x_k) \leq 0, \quad \forall \{i, j, k\} \subseteq V, \quad i < j < k, \\
& 2(-x_i - x_j x_k + x_i x_j + x_i x_k) \leq 0, \quad \forall \{i, j, k\} \subseteq V, \quad i < j < k, \\
& 2(-x_j - x_i x_k + x_i x_j + x_j x_k) \leq 0, \quad \forall \{i, j, k\} \subseteq V, \quad i < j < k, \\
& -x_i \leq 0, \quad x_i - 1 \leq 0, \quad \forall i \in V.
\end{aligned} \tag{61}$$

Aplicando a (61) a suavização hiperbólica, obtemos a seguinte formulação irrestrita, chamada de (MC_θ^ϕ) :

$$\min - \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} \theta(x_i - x_j, \gamma) + \sum_{i < j < k}^{n-2} \phi(\lambda, \tau, f_{ijk}(x)) + \sum_{i < j < k, l=1}^3 \phi(\lambda, \tau, g_{ijk}^l(x)) + \sum_{i=1}^n \sum_{l=0}^1 \phi(\lambda, \tau, h_i^l(x)) \tag{62}$$

onde

$$\begin{aligned}
f_{ijk} &= 2(x_i + x_j + x_k) - 2(x_i x_j + x_i x_k + x_j x_k) - 2, \\
g_{ijk}^1 &= -2x_k - 2x_i x_j + 2x_i x_k + 2x_j x_k, \\
g_{ijk}^2 &= -2x_i - 2x_j x_k + 2x_i x_j + 2x_i x_k, \\
g_{ijk}^3 &= -2x_j - 2x_i x_k + 2x_i x_j + 2x_j x_k,
\end{aligned}$$

$$\begin{aligned}
h_i^0 &= -x_i, \\
h_i^1 &= 1 - x_i, \\
\theta(u, \gamma) &= \sqrt{u^2 + \gamma^2}, \\
\phi(\lambda, \tau, y) &= \lambda y + \sqrt{\lambda^2 y^2 + \tau^2}.
\end{aligned}$$

A solução do problema (62) retorna um ponto $x \in [0, 1]^n$. Esse ponto pode ser convertido numa solução para o Max-Cut de diferentes formas, sendo a mais simples via arredondamento. O Algoritmo 1 esboça uma possibilidade de uso do problema suavizado para gerar uma solução viável do Max-Cut.

Algoritmo 1: Heurística Baseada na Suavização Hiperbólica

```

Escolha  $nStart$  ;                               /* Número de reinicializações */
 $fBestInt \leftarrow -\infty$ ;
para  $k = 1, \dots, nStart$  faça
  se  $k = 1$  então
    |  $x \leftarrow$  ponto aleatório;
    |  $xBest \leftarrow x$ ;
  senão
    |  $x \leftarrow$  Perturbação de  $xBest$ ;
  fim
  Escolha  $maxIt$  ;                               /* max.número iterações do BFGS */
  Escolha  $\gamma, \tau$  e  $\lambda$  ;                 /* Valores iniciais dos parâmetros */
  Escolha  $r_\gamma, r_\tau$  e  $r_\lambda$  ;           /* fatores de atualização dos parâmetros */
   $nIt \leftarrow 0$ ;
  repita
    |  $nIt \leftarrow nIt + 1$ ;
    |  $x \leftarrow$  BFGS( $x$ ) ;                   /* BFGS Quasi Newton a partir de  $x$  */
    |  $fSuave \leftarrow$  ValorFuncaoObjetivoSuave( $x$ ) ; /* Valor minimização */
    |  $xInt \leftarrow$  Converte01( $x$ ) ;          /* Converte solução fracionária em
    |   inteira */
    |  $fInt \leftarrow$  ValordoCorte( $xInt$ ) ;     /* Calcula valor do corte definido
    |   por  $xInt$  */
    | se  $fInt > fBestInt$  então
    |   |  $fBestInt \leftarrow fInt$ ;
    |   |  $xBestInt \leftarrow xInt$ ;
    |   |  $xBest \leftarrow x$ ;
    | fim
    |  $\gamma \leftarrow \gamma/r_\gamma, \tau \leftarrow \tau/r_\tau, \lambda \leftarrow r_\lambda \lambda$ ;
  até  $nIt \geq maxIt$  ou  $|fInt + fSuave|/|fInt| < 0.001$ ;
fim
return  $xBestInt$ ;

```

3.7 Heurísticas para o Max-Cut

3.7.1 Breve revisão bibliográfica

Devido à complexidade de resolver exatamente problemas difíceis, como é o caso do Max-Cut, vários pesquisadores têm investido em técnicas híbridas que usam heurísticas para obter bons limites superiores e inferiores em tempo razoável.

O trabalho de Goemans e Williamson (1995) garante solução com uma boa qualidade de aproximação da solução ótima para instâncias do problema do Max-cut onde os pesos das arestas são não negativos. No entanto, o custo computacional associado a essa boa aproximação é elevado para problemas de grande escala, devido a necessidade de resolver relaxação da programação semidefinida. Diante desse fato, surge a seguinte pergunta: Pode a abordagem inspirada por Goemans e Williamson (1995), que se fundamenta na resolução da relaxação PDS, chegar a ser competitiva para problemas de grande escala? O método CirCut, desenvolvido por Burer, Monteiro, e Zhang (2002), é uma modificação daquele apresentado no trabalho de Goemans e Williamson. Nessa nova heurística, eles relaxam o vetor binário em um vetor de ângulos e trabalham com uma representação angular do corte. O trabalho foi comparado com outros dois métodos que usam o algoritmo de Goemans e Williamson. Os resultados indicam que CirCut retorna soluções aproximadas com boa qualidade e em pouca quantidade de tempo, sendo assim indicado para resolver situações onde o problema é muito grande e há pouca disponibilidade de tempo.

Em Festa *et al.* (2002), os autores desenvolveram seis heurísticas para o Max-Cut, sendo as duas primeiras GRASP e *Variable Neighborhood Search* (VNS), enquanto as quatro últimas são formas híbridas das duas primeiras juntamente com *Path-Relinking* (PR): GRASP-PR, GRASP-VNS, VNS-PR e GRASP-VNS-PR. Os autores usam um conjunto de testes (24 instâncias) criados por Helmberg e Rendl (2000), que usaram um gerador de grafos escrito por Rinaldi (1998). A ideia inicial era comparar as heurísticas com o trabalho de Goemans e Williamson, mas Burer et al (2001) mostraram que CirCut produz melhores soluções aproximadas na prática do que Goemans e Williamson. Os resultados mostraram que, na comparação entre CirCut e VNS-PR, CirCut encontrou melhores soluções em 13 das 24 instâncias, 4 das 24 foram empates e VNS-PR ganhou em 7. Para todos os casos CirCut encontrou a solução em menor tempo.

No trabalho de Alperin e Nowak (2005), os autores propõem uma heurística que chamaram *Lagrangian Smoothing Truncated Projected Gradient* (LS-TPG), que também é chamado de deformação, ou suavização heurística. A heurística é baseada em um problema de otimização paramétrica, definido como uma combinação convexa entre a relaxação lagrangiana e o problema original. A partir da relaxação lagrangiana, o método *pathfollowing* é aplicado para obter boas soluções, enquanto gradualmente o problema relaxado é transformado no problema original, formulado com uma função de penalidade exata. A heurística LS-TPG foi comparada com o trabalho de Goemans e Williamson, GRASP-VNS e CirCut. Os resultados obtidos mostram que LS-TPG não é tão bom quanto CirCut, mas possui desempenho semelhante ao GRASP-VNS. Os autores afirmam que é a primeira vez que uma heurística lagrangiana é combinada com técnicas de suavização.

No trabalho de Xia e Xu (2010), os autores propõem uma heurística mais efi-

cientista denominado de *Lagrangian Smoothing - Truncated Frank-Wolfe* (LS-TFW). Essa heurística é uma versão melhorada da LS-TPG, onde os subproblemas contínuos resultantes das funções suavizadas são resolvidos pelo algoritmo de Frank-Wolfe truncado. A heurística LS-TFW foi comparada com a heurística LS-TPG e com o trabalho CirCut de Burer et al(2001), usando um conjunto de teste criado por Rudy (gerador de instâncias desenvolvido por Giovanni Rinaldi). Os resultados mostraram que o LS-TFW obteve melhores soluções em menos tempo que o LS-TPG. Com relação ao trabalho de Burer et al(2001), o LS-TFW obteve soluções em menos tempo, mas apenas 30% delas eram melhores.

Em Martí, Duarte, e Laguna (2009), os autores propõem um método heurístico baseado na metodologia *scatter-search* (SS) para encontrar soluções aproximadas para o problema do Max-Cut. O procedimento incorpora algumas características inovadoras, no âmbito da *scatter-search*: (1) a solução do problema da diversidade máxima para aumentar a diversidade no conjunto de referência, (2) um ajuste dinâmico de um parâmetro-chave dentro da pesquisa, e (3) a seleção adaptativa de um método de combinação. Os autores montaram 3 conjuntos de testes e comparam a eficiência do método com a eficiência de alguns métodos da literatura, CirCut e VNS-PR. Em alguns casos, SS mostrou-se mais eficiente do que CirCut e, comparado a VNS-PR, mostrou-se ainda melhor.

Embora populares, os métodos desenvolvidos em Burer, Monteiro, e Zhang (2002) e Martí, Duarte, e Laguna (2009) foram superados pelos métodos descritos em Benlic e Hao (2013), Shylo, Shylo, e Roschyn (2012) e Wang *et al.* (2012), que, por sua vez, foram superados por Wu, Wang, e Lü (2015). Como base nos resultados obtidos pelos trabalhos pesquisados, fizemos uma análise levando em consideração o tempo gasto por cada método heurístico para obter o valor aproximado do corte máximo para determinado problema.

Feita essa análise, concluímos que os melhores métodos são em geral bem complexos e de difícil implementação. A partir dessa observação e considerando que desejamos uma heurística para incorporar em um método exato, decidimos por desenvolver heurísticas simples, além daquela baseada na suavização hiperbólica.

3.7.2 Heurística gulosa com busca local

A primeira heurística que propomos depende apenas da análise dos custos das arestas. Ela é composta por duas fases: a primeira fase é um algoritmo guloso, e a segunda aplica um algoritmo de busca local. O pseudocódigo de cada fase pode ser visto nos algoritmos 2 e 3, respectivamente.

Na primeira fase, um corte (S, \bar{S}) é construído iterativamente. Os conjuntos S e \bar{S} começam cada um com uma das extremidades da aresta de maior custo. Dado o par (S, \bar{S}) corrente, a contribuição de um vértice $v \in V \setminus (S \cup \bar{S})$ é $\max\{\sum_{a \in S} c_{va}, \sum_{b \in \bar{S}} c_{vb}\}$.

A cada iteração, escolhe-se o vértice v que mais contribui para aumentar o peso do corte, colocando-o em S , se $\sum_{a \in S} c_{va} \geq \sum_{b \in \bar{S}} c_{vb}$, ou em \bar{S} , caso contrário.

A segunda fase é composta de três partes. Na primeira parte, para cada par $(i, j) \in S \times \bar{S}$, compara-se o valor do corte (S, \bar{S}) com o valor do corte $(S \setminus \{i\} \cup \{j\}, \bar{S} \setminus \{j\} \cup \{i\})$ e atualiza-se (S, \bar{S}) , caso o segundo tenha maior valor. Na segunda parte, para cada $i \in S$, compara-se o valor do corte (S, \bar{S}) com o valor do corte $(S \setminus \{i\}, \bar{S} \cup \{i\})$ e atualiza-se (S, \bar{S}) , caso o segundo tenha maior valor. Por fim, na terceira parte, para cada $i \in \bar{S}$, compara-se o valor do corte (S, \bar{S}) com o valor do corte $(S \cup \{i\}, \bar{S} \setminus \{i\})$ e atualiza-se (S, \bar{S}) , caso o segundo tenha maior valor. Essas trocas são avaliadas, até que não se consiga mais obter uma melhor partição (S, \bar{S}) .

Algoritmo 2: Heurística Gulosa

Entrada: Grafo completo $G = (V, E)$, função de pesos $c : E \rightarrow \mathbb{R}$

Saída: (S, \bar{S})

início

$c_{i^*j^*} = \max\{c_{ij} : i, j \in \{1, \dots, n\}\}$, $S \leftarrow \{i^*\}$, $\bar{S} \leftarrow \{j^*\}$ e $V' \leftarrow V \setminus \{i, j\}$;

enquanto $V' \neq \emptyset$ **faça**

$P_{max} \leftarrow -\infty$;

para cada $v \in V'$ **faça**

$P_a \leftarrow \sum_{a \in S} c_{va}$, $P_b \leftarrow \sum_{b \in \bar{S}} c_{vb}$;

se $\max\{P_a, P_b\} > P_{max}$ **então**

se $P_a \geq P_b$ **então**

$v_{max} \leftarrow v$, $parte \leftarrow 1$, $P_{max} \leftarrow P_a$;

senão

$v_{max} \leftarrow v$, $parte \leftarrow 0$, $P_{max} \leftarrow P_b$;

fim

fim

fim

se $parte = 1$ **então**

$S \leftarrow S \cup \{v_{max}\}$;

senão

$\bar{S} \leftarrow \bar{S} \cup \{v_{max}\}$;

fim

$V' \leftarrow V' \setminus \{v_{max}\}$;

fim

 Busca Local(S, \bar{S});

fim

return (S, \bar{S}) ;

Algoritmo 3: Busca Local

Entrada: (S, \bar{S}) **Saída:** (S, \bar{S}) **início** **repita** **repita** $melhorValor \leftarrow 0;$ **para cada** $i \in S$ e $j \in \bar{S}$ **faça** $A \leftarrow S \setminus \{i\} \cup \{j\}, B \leftarrow \bar{S} \setminus \{j\} \cup \{i\};$ **se** $valor_corte(A, B) > valor_corte(S, \bar{S})$ **então** $S \leftarrow S \setminus \{i\} \cup j, \bar{S} \leftarrow \bar{S} \setminus \{j\} \cup \{i\};$ $melhorParticao \leftarrow 1, melhorValor \leftarrow 1;$ **fim** **fim** **até** $melhorValor = 0;$ **repita** $melhorValor \leftarrow 0;$ **para cada** $i \in S$ **faça** $A \leftarrow S \setminus \{i\}, B \leftarrow \bar{S} \cup \{i\};$ **se** $valor_corte(A, B) > valor_corte(S, \bar{S})$ **então** $S \leftarrow S \setminus \{i\}, \bar{S} \leftarrow \bar{S} \cup \{i\};$ $melhorParticao \leftarrow 1, melhorValor \leftarrow 1;$ **fim** **fim** **até** $melhorValor = 0;$ **repita** $melhorValor \leftarrow 0;$ **para cada** $i \in \bar{S}$ **faça** $A \leftarrow S \cup \{i\}, B \leftarrow \bar{S} \setminus \{i\};$ **se** $valor_corte(A, B) > valor_corte(S, \bar{S})$ **então** $S \leftarrow S \cup \{i\}, \bar{S} \leftarrow \bar{S} \setminus \{i\};$ $melhorParticao \leftarrow 1, melhorValor \leftarrow 1;$ **fim** **fim** **até** $melhorValor = 0;$ **até** $melhorParticao = 0;$ **fim****return** $(S, \bar{S});$

3.7.3 Heurística baseada em condições de otimalidade

A segunda heurística, também gulosa, é fundamentada nas propriedades estabelecidas na Subseção 3.2.2. Particularmente, consideramos a condição (34), que determina a fixação de variáveis.

Dada uma solução parcial \hat{x} e $k \in J_2(\hat{x})$, seja

$$\Delta(k, \hat{x}) = C_2(k, \hat{x}) - |C_{01}(k, \hat{x})|,$$

onde

$$C_2(k, \hat{x}) = \sum_{j \in J_2(\hat{x})} |c_{kj}| \quad \text{e} \quad C_{01}(k, \hat{x}) = \sum_{j \in J_0(\hat{x}) \cup J_1(\hat{x})} (1 - 2\hat{x}_j)c_{kj},$$

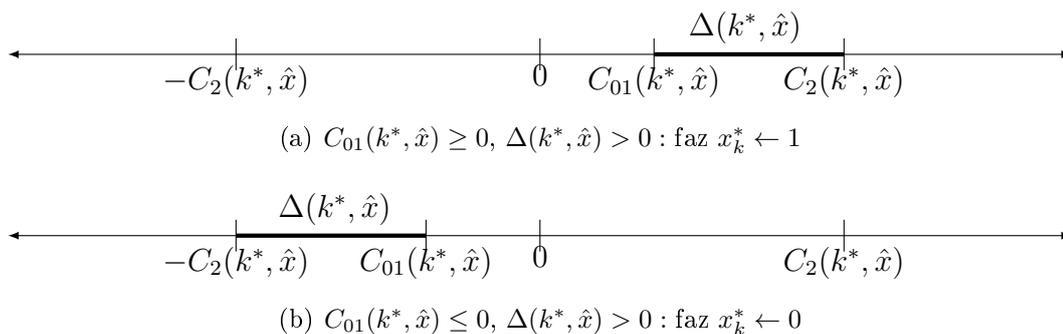
conforme definidos em (35).

Se $\Delta(k, \hat{x}) \leq 0$, podemos fixar a variável x_k . Como determinam os teoremas 3.2 e 3.3, podemos fazer $x_k = 1$ (se $C_{01}(k, \hat{x}) \geq 0$) ou $x_k = 0$ (se $C_{01}(k, \hat{x}) \leq 0$). Ver Figura 9. Caso exista algum k tal que $\Delta(k, \hat{x}) > 0$, podemos determinar a variável x_{k^*} mais propensa a ter seu valor fixado, conforme esse mesmo raciocínio. Ela é tal que

$$k^* \in \arg \min \{ \Delta(k, \hat{x}) : k \in J_2(\hat{x}), \Delta(k, \hat{x}) > 0 \}$$

Dessa forma, escolhemos fazer $x_{k^*}^* = 1$ (se $C_{01}(k^*, \hat{x}) \geq 0$) e $x_{k^*}^* = 0$ (se $C_{01}(k^*, \hat{x}) < 0$), ver Figura 10.

Figura 10: Decisão heurística. Casos em que $\Delta(k^*, \hat{x}) > 0$, ou seja, $|C_{01}(k^*, \hat{x})| < C_2(k^*, \hat{x})$.



Fonte: Elaborado pelo autor.

Como na primeira heurística, construímos o corte iterativamente, usando as variáveis $x_i, i \in \{1, \dots, n\}$, para identificar o conjunto $S = \{i : x_i = 1\}$ e $\bar{S} = \{i : x_i = 0\}$. Ao longo do processo, vamos construindo uma sequência de soluções parciais, até obter uma solução completa, usando a fixação de variáveis dadas pelas regras descritas acima. Um pseudocódigo dessa heurística pode ser visto no Algoritmo 4. Como solução parcial inicial tomamos $x_{i^*} = 1$ e $x_{j^*} = 0$, onde $c_{i^*j^*} = \max\{c_{ij} : i, j \in \{1, \dots, n\}\}$.

Algoritmo 4: Heurística baseada em otimalidade

Entrada: Solução parcial x , função de custos $c : E \rightarrow \mathbb{R}$;
Saída: Solução viável x ;
 $J_{01} = \{j : x_j \in \{0, 1\}\}$;
 $J_2 = \{1, \dots, n\} \setminus J_{01}$;
 $C_2(k) \leftarrow \sum_{j \in J_2} |c_{kj}|$, $C_{01}(k) \leftarrow \sum_{j \in J_{01}} (1 - 2x_j)c_{kj}$, $\forall k \in J_2$;
enquanto $J_2 \neq \emptyset$ **faça**
 $\Delta min \leftarrow \infty$;
 para cada $k \in J_2$ **faça**
 $\Delta k \leftarrow C_2(k) - |C_{01}(k)|$;
 se $\Delta k \leq 0$ **então**
 se $C_{01}(k) \geq C_2(k)$ **então** $x_k \leftarrow 1$ **senão** $x_k \leftarrow 0$;
 $J_2 \leftarrow J_2 \setminus \{k\}$;
 $C_2(\ell) \leftarrow C_2(\ell) - |c_{k\ell}|$, $C_{01}(\ell) \leftarrow C_{01}(\ell) + (1 - 2x_k)c_{k\ell}$, $\forall \ell \in J_2$;
 senão
 se $\Delta min < \Delta k$ **então**
 $\Delta min \leftarrow \Delta k$;
 $k^* \leftarrow k$;
 fim
 fim
 se $\Delta min < \infty$ **então**
 se $C_{01}(k^*) \geq 0$ **então** $x_{k^*} \leftarrow 1$ **senão** $x_{k^*} \leftarrow 0$;
 $J_2 \leftarrow J_2 \setminus \{k^*\}$;
 $C_2(\ell) \leftarrow C_2(\ell) - |c_{k^*\ell}|$, $C_{01}(\ell) \leftarrow C_{01}(\ell) + (1 - 2x_{k^*})c_{k^*\ell}$, $\forall \ell \in J_2$;
 fim
fim
return x ;

3.8 Experimentos Computacionais e Comentários

Apresentamos nesta seção três experimentos computacionais. No experimento 1, vamos observar a qualidade do limite superior obtido pela t -linearização, quando comparado com as linearizações clássica e de Glover. Pelos teoremas 3.4 e 3.5, sabemos que os três limites são iguais para o problema Max-Cut. Nossa intenção é averiguar o tamanho do modelo linear (número de restrições) e o esforço computacional demandado em cada caso, lembrando que a t -linearização requer um processo de separação de restrições. Já no experimento 2, vamos observar a qualidade dos limites inferiores obtidos por três variações da heurística derivada com a suavização hiperbólica (Algoritmo 1) e pelas heurísticas propostas (algoritmos 2 e 4). E por fim, no experimento 3 estamos interessados em testar a força das desigualdades válidas apresentadas na Seção 3.2. Utilizamos a formulação (60) como base para os experimentos com a t -linearização, apenas com as variáveis x , sem as variáveis complementares \bar{x} .

A máquina utilizada para executar os experimentos computacionais possui a seguinte configuração: Processador Intel[®] Core[™] i5-4570 com 3.20 GHz, 16 GB RAM. As instâncias utilizadas nos 2 primeiros experimentos, descritas a seguir, foram geradas por **rudy** (Rinaldi, 1998) e estão disponíveis em <http://biqmac.uni-klu.ac.at/biqmaclib.html> e <http://www.opticom.es/maxcut>. As instâncias do conjunto 3 foram geradas por nós.

Conjunto 1 : 130 Instâncias, sendo 10 instâncias para cada combinação de n (nº de vértices) e d (densidade de arestas com pesos não nulos)

- **g05_n.i**: Grafos com $n \in \{60, 80, 100\}$ vértices e probabilidade 0.5 de existência de cada aresta (não ponderada), totalizando 30 instâncias.
- **pm1s_n.i** e **pm1d_n.i**: Grafos completos com $n \in \{80, 100\}$ vértices, pesos das arestas escolhidos uniformemente entre $\{-1, 0, 1\}$, e $d = 10\%$ ou $d = 99\%$ respectivamente, totalizando 40 instâncias.
- **wd_n.i**: Grafos completos com $n = 100$ vértices, pesos (inteiros) das arestas escolhidos em $[-10, 10]$ e $d \in \{0.1, 0.5, 0.9\}$, totalizando 30 instâncias.
- **pwd_n.i**: Grafos completos com $n = 100$ vértices, pesos (inteiros) das arestas escolhidos em $[0, 10]$ e $d \in \{0.1, 0.5, 0.9\}$, totalizando 30 instâncias.

Em cada grupo, i identifica uma instância em particular.

Conjunto 2 : 54 instâncias que consistem em grafos planares, toroidais, com pesos nas arestas em $\{-1, 0, 1\}$ e $n \in \{800, 1000, 2000, 3000\}$.

Conjunto 3 : 60 instâncias, sendo 5 instâncias para cada combinação de $n \in \{30, 50, 70, 100\}$ e $d \in \{0.1, 0.25, 0.5\}$, com arestas ponderadas no intervalo $[-100, 100]$.

3.8.1 Comparação entre os limites superiores

A qualidade do limite superior dado pela t -linearização é comparada à qualidade do limite superior dado pelas linearizações Clássica e Glover. As três formulações foram implementadas usando o software comercial IBM/ILOG CPLEX 12.7 em conjunto com Ambiente de Desenvolvimento Integrado (IDE - Integrated Development Environment) *Code::Blocks* utilizando a linguagem C++ e sistema operacional Ubuntu 16.04 LTS. Para essa comparação usamos as 130 instâncias do conjunto 1.

A Tabela 5 mostra, para cada subgrupo de instâncias do conjunto 1, as médias do desvio percentual (gap), do tempo (CPU), da quantidade de restrições (R_f) para as formulações clássica e Glover, e do número de restrições geradas (N_{rg}) pela t -linearização. O gap foi calculado de acordo com a seguinte fórmula

$$gap = \frac{LS - VO}{VO} \times 100,$$

onde LS é o limite superior fornecido pela linearização e VO é o valor da solução ótima. É importante notar que o procedimento de linearização clássica gera 3 restrições para cada par $\{i, j\} \in E, i \neq j$, no entanto podemos descartar as restrições do tipo $y_{ij} \geq x_i + x_j - 1$

quando os pesos das arestas são não negativos. Cada linha da Tabela 5, a partir da terceira, apresenta a média (gap , CPU, R_f e N_{rg}) das 10 instâncias daquele subgrupo.

Tabela 5: CPU, gap , R_f e N_{rg} obtidos pela t -linearização, linearização clássica e linearização Glover

	n	Linearização Clássica			Linearização Glover			t-Linearização		
		gap	CPU	R_f	gap	CPU	R_f	gap	CPU	N_{rg}
g05	60	67.79	0.005	3540	67.79	0.004	118	67.79	0.032	61
g05	80	71.64	0.012	6320	71.64	0.010	158	71.64	0.091	81
g05	100	74.65	0.021	9900	74.65	0.016	198	74.65	0.219	101
pm1s	80	122.46	0.012	9480	122.46	0.006	158	122.46	0.068	81.2
pm1s	100	128.67	0.019	14850	128.67	0.010	198	128.67	0.164	101
pm1d	80	566.63	0.011	9480	566.63	0.010	158	566.63	0.080	81
pm1d	100	597.82	0.023	14850	597.82	0.019	198	597.82	0.183	101.1
w01	100	102.61	0.020	14850	102.61	0.013	198	102.61	0.149	101
w05	100	319.47	0.023	14850	319.47	0.017	198	319.47	0.175	101
w09	100	488.62	0.021	14850	488.62	0.022	198	488.62	0.180	101
pw01	100	36.70	0.018	9900	36.70	0.012	198	36.70	0.159	101.7
pw05	100	69.54	0.017	9900	69.54	0.016	198	69.54	0.214	101
pw09	100	82.10	0.021	9900	82.10	0.018	198	82.10	0.227	101
<i>Média</i>		209.90	0.017	10974.61	209.90	0.013	182.61	209.90	0.149	93.38

Fonte: Elaborado pelo autor.

Os resultados obtidos reafirmam que os gaps obtidos pela t -linearização, linearização clássica e linearização Glover são iguais, como já estabelecia os teoremas 3.4 e 3.5. Com relação ao tempo, a linearização Glover e clássica demandaram tempo semelhante na obtenção dos limites, enquanto que a t -linearização fez uso de um pouco mais. Com relação à quantidade de restrições utilizadas, podemos ver que, apesar da igualdade na obtenção dos limites, a t -linearização utilizou um número de restrições muito menor, comparada ao número total de partições e permutações possíveis ($2^n \cdot n!$), assim como das $O(n^2)$ restrições utilizadas pela linearização clássica e $O(n)$ da linearização Glover.

3.8.2 Comparação entre os limites inferiores

A qualidade dos limites inferiores dados pelas heurísticas gulosas das subseções 3.7.2 e 3.7.3 é comparada com aquela dada por três versões da suavização hiperbólica aplicada ao Max-Cut. A primeira versão utiliza a formulação (62), descrita na Seção 3.6, e será denominada aqui de $SH + \Delta$. O resultado dessa versão é, portanto, a aplicação da Heurística 1 com o arrendamento da solução fracionária obtida por (62). A segunda versão é uma variação da primeira, denominada SH , onde para este caso não consideramos as restrições triangulares em (62). A terceira versão, denominada aqui $SH + \Delta + BL$, é uma junção de $SH + \Delta$ com o procedimento de busca local descrito no algoritmo 3.

As versões foram implementadas usando o software Microsoft Visual Studio em conjunto com Intel[®] Visual Fortran Composer XE 2017 Update 1 e sistema operacional Windows 7. O código da versão SH foi gentilmente disponibilizado pelo Professor Adilson Xavier. A formulação suavizada utiliza um dos métodos iterativos Quasi-Newton,

denominado BFGS (Broyden-Fletcher-Goldfarb-Shanno). Para mais detalhes sobre o funcionamento do BFGS, ver Dai (2013). Os parâmetros usados nas três versões foram $\gamma = 0.99$, $\tau = 0.09$ e $\lambda = 40 \times \max\{|c_{ij}|, \forall(i, j) \in E\}$, $nStart = 25$, $maxit = 6$, $r_\lambda = 1$ e $r_\gamma = r_\tau = 1.5$.

Tabela 6: LB e CPU obtidos pelas heurísticas gulosas, SH , $SH + \Delta$ e $SH + \Delta + BL$

Inst	V	H Gulosa + BL		H Otimalidade		SH		$SH + \Delta$		$SH + \Delta + BL$	
		LB	CPU	LB	CPU	LB	CPU	LB	CPU	LB	CPU
G_1	800	11446	16.29	11232	0.02	10522	11.07	10422	7.79	11536	424.3
G_2	800	11451	17.18	11213	0.02	10587	12.04	10462	7.66	11536	2247.79
G_3	800	11413	16.31	11247	0.02	10383	11.79	10121	7.62	11540	124.69
G_4	800	11406	21.65	11176	0.02	10490	11.75	10504	7.72	11550	2138
G_5	800	11406	17.97	11230	0.02	10538	12.76	10488	7.72	11573	173.56
G_6	800	2073	11.48	1840	0.02	1999	40.07	2014	19.94	2054	2437.54
G_7	800	1834	11.07	1688	0.02	1797	43.43	1871	19.34	1900	458.59
G_8	800	1846	10.62	1562	0.02	1882	43.83	1842	19.33	1958	709.71
G_9	800	1876	10.33	1569	0.02	1920	41.8	1794	19.19	1961	304.5
G_{10}	800	1804	8.99	1493	0.02	1864	44.44	1919	20.41	1926	376.18
G_{11}	800	502	1.27	450	0.02	522	33.24	520	24.61	524	568.68
G_{12}	800	492	1.92	426	0.02	518	33.8	524	23.24	526	471.13
G_{13}	800	536	1.27	428	0.02	554	33.16	554	23.69	560	917.95
G_{14}	800	3012	4.35	2865	0.02	0	3.18	0	2.9	2992	1219.62
G_{15}	800	2977	7.11	2845	0.02	0	1.76	0	1.97	2968	1240.61
G_{16}	800	2968	4.94	2836	0.02	0	3.33	0	2.92	2974	1020.67
G_{17}	800	2961	6	2846	0.02	0	3.45	0	2.92	2964	6555.6
G_{18}	800	924	4.53	757	0.02	929	37.37	940	24.24	948	849.04
G_{19}	800	828	4.14	650	0.02	857	40.5	824	22.73	862	817.43
G_{20}	800	842	4.89	739	0.02	871	37.47	884	22.11	912	944.09
G_{21}	800	841	4.27	705	0.02	868	40.71	888	23.29	896	492.92
G_{22}	2000	13057	514.07	12654	0.08	12322	327.29	11813	150.81	13132	10361.1
G_{23}	2000	13064	424.56	12795	0.08	12446	309.73	11790	150.56	13165	2205.14
G_{24}	2000	13017	483.22	12642	0.08	12341	318.39	11717	150.51	13087	2791.86
G_{25}	2000	13046	495.09	12679	0.08	12241	302.1	11749	149.6	13084	1757.76
G_{26}	2000	12996	462.92	12739	0.08	12139	290.29	11816	150.01	13096	1705.3
G_{27}	2000	3073	421.63	2698	0.08	3080	353.83	3073	305.86	3165	1794.46
G_{28}	2000	3018	236.91	2651	0.08	3075	345.73	3075	290.64	3108	492.35
G_{29}	2000	3086	206.55	2721	0.08	3142	368.41	3138	309.74	3226	577.42
G_{30}	2000	3110	307.68	2819	0.08	3182	364.47	3170	316.69	3198	1081.24
G_{31}	2000	3043	162.95	2580	0.08	3093	375.23	3025	312.09	3144	2964.28
G_{32}	2000	1254	38.47	1074	0.06	1312	349.81	1294	360.72	1308	1754.78
G_{33}	2000	1260	38.07	1038	0.06	1280	349.18	1244	370.08	1292	1691.91
G_{34}	2000	1260	38.88	994	0.06	1306	353.63	1288	364.76	1288	2075.67
G_{35}	2000	7501	301.59	7203	0.07	0	2.08	0	2.10	7041	2327.55
G_{36}	2000	7481	434.64	7176	0.07	0	1.58	0	2.02	6943	2498.88
G_{37}	2000	7510	444.98	7166	0.07	0	1.45	0	1.50	7000	2838.15
G_{38}	2000	7518	457.77	7163	0.07	0	1.79	0	1.82	6978	2621.34
G_{39}	2000	2197	139.28	1811	0.07	2232	367.33	2105	347.78	2307	1998.43
G_{40}	2000	2193	199.10	1785	0.07	2207	363.24	2221	381.56	2269	3339.98
G_{41}	2000	2174	157.72	1800	0.07	2236	358.76	2243	359.05	2278	1626.97
G_{42}	2000	2327	210.86	1852	0.07	2302	725.69	2271	356.58	2356	2833.5
G_{43}	1000	6511	26.28	6400	0.2	6025	24.25	5878	15.75	6529	54.72
G_{44}	1000	6542	27.55	6368	0.2	5989	24.25	5991	15.73	6550	244.51
G_{45}	1000	6519	24.96	6308	0.2	5969	24.27	5887	15.96	6552	83.14
G_{46}	1000	6537	23.38	6326	0.2	5973	25.25	5984	15.78	6529	47.94
G_{47}	1000	6524	30.08	6321	0.2	6031	25.78	5928	15.70	6563	374.51
G_{48}	3000	6000	1339.87	6000	0.2	5446	888.74	4920	519.56	6000	5565.84
G_{49}	3000	6000	2915.13	6000	0.2	5488	920.1	5018	517.37	6000	1919.09
G_{50}	3000	5880	975.21	5880	0.2	5570	882.01	5038	551.34	5874	2651.82
G_{51}	1000	3764	14.02	3611	0.02	0	1.8	0	2.64	3527	218.17
G_{52}	1000	3762	11.01	3582	0.02	0	1.9	0	1.80	3511	260.84
G_{53}	1000	3756	16.29	3620	0.02	0	2.65	0	5.07	3494	250.05
G_{54}	1000	3767	13.31	3614	0.02	0	4.73	0	4.99	3528	213.68
Média			414		0.04		177.71		126.43		1624.34

Fonte: Elaborado pelo autor.

A Tabela 6 mostra o limite inferior, LB , encontrado por cada método e o tempo médio, CPU, correspondente. Destacamos em negrito os melhores resultados, conforme descrição dos experimentos. Podemos observar que a versão $SH + \Delta + BL$ obteve melhores limites, dentre os cinco métodos testados, em 77% das instâncias (42 de 54). Por outro lado, essa versão foi a que utilizou mais tempo de execução. Com relação às heurísticas gulosas, observa-se que a heurística baseada nos critérios de otimalidade é muito rápida, mas a heurística gulosa que usa busca local obtém limites melhores. Vale observar ainda, na Tabela 6, que as heurísticas SH e $SH + \Delta$ retornaram limite inferior nulo para 12 instâncias. Isto porque o ponto retornado pela suavização hiperbólica tinha todas as coordenadas abaixo de 0.5. Uma explicação para esse resultado encontra-se no fato de que essas instâncias estão entre aquelas que possuem menor densidade de arestas não nulas (em torno de 1.07%) e no fato de os custos das demais arestas serem iguais a 1.

3.8.3 Força das desigualdades

No experimento 3, comparamos implementações da formulação linear inteira (60), obtidas quando são consideradas as seguintes opções: (i) as restrições t -linearizadas são incluídas local ou globalmente ao longo da árvore de busca, (ii) as desigualdades válidas (38)-(39), que foram apresentadas na Subseção 3.2.3, são ou não adicionadas para fortalecer a formulação.

Nas implmentações, usamos o software comercial IBM/ILOG CPLEX 12.7 integrado com Ambiente de Desenvolvimento Integrado (IDE - Integrated Development Environment) *Code::Blocks* utilizando a linguagem C++ e sistema operacional Ubuntu 16.04 LTS. Utilizamos as instâncias do conjunto 3 e limitamos o tempo computacional do CPLEX em 1h (3600s).

Em todas as versões, a primeira restrição da t -linearização é gerada a partir de uma solução fornecida por uma heurística. Essa restrição é adicionada ao modelo, que em seguida é resolvido pelo algoritmo de branch-and-bound. O procedimento de separação das desigualdades é usado dentro do branch-and-bound, sempre que uma solução inteira é encontrada, ou seja, as restrições da t -linearização são usadas como *lazy constraints*.

A partir dessa implementação base, avaliamos três variações. A primeira, referenciada aqui por $(FL)_t$, não emprega as desigualdades (38)-(39) e aplica as restrições da t -linearização globalmente, ou seja, uma vez adicionadas, elas são usadas em todos os nós processados após sua inclusão. As outras duas versões, denominadas $(FL)_{tG}RV$ e $(FL)_{tL}RV$, adicionam as desigualdades válidas (38)-(39) na raiz da árvore de busca. Todavia, enquanto $(FL)_{tG}RV$, assim como $(FL)_t$, aplica as restrições da t -linearização globalmente, $(FL)_{tL}RV$ as considera apenas localmente, ou seja, quando adicionada em um nó, uma restrição é mantida apenas nos nós descendentes.

Para cada instância calculamos o $gap = \left[\frac{UB-LB}{LB} \times 100 \right]$, onde UB e LB são os limites inferiores e superiores respectivamente fornecidos pelo CPLEX. As tabelas 7

e 8, mostram a partir da linha 3, a média do gap, GAP, do tempo de computação em segundos, CPU, da quantidade de restrições geradas pelo procedimento de separação da t -linearização, N_{rg} , assim como a quantidade de soluções ótimas encontradas por cada formulação, #Opt. Destacamos em negrito os melhores resultados, conforme descrição dos experimentos.

Quando comparamos $(FL)_t$ com $(FL)_{tGRV}$, os resultados obtidos mostram que, quando adicionamos as desigualdades válidas da Subseção 3.2.3, o número de restrições geradas pela t -linearização diminuiu consideravelmente em todas as combinações de (n, d) , assim como o gap médio. O número de soluções ótimas obtidas também melhorou, passando de 23%(14 de 60) para 45%(27 de 60). Já quando comparamos $(FL)_{tGRV}$ com $(FL)_{tLRV}$, percebemos que o número de soluções ótimas aumentou de 45%(27 de 60) para 58.33%(35 de 60). Por outro lado, como era esperado, a quantidade de restrições geradas pela t -linearização aumentou bastante na versão $(FL)_{tLRV}$. Esse fenômeno indica que uma mesma restrição é gerada mais de uma vez na árvore de busca. Mesmo assim, o tempo de computação médio da versão com restrições locais foi 15% menor que aquele com as restrições globais.

Vale destacar que também implementamos uma versão de $(FL)_t$ com adição das restrições da t -linearização localmente. No entanto, a quantidade de restrições geradas em cada instância, a partir das combinações de $n = 30$ e $d = 25$, excedeu a capacidade de memória RAM da máquina, levando ao travamento da mesma.

Tabela 7: Comparação entre $(FL)_t$ e $(FL)_{tGRV}$

n	d	$(FL)_t$				$(FL)_{tGRV}$			
		GAP	CPU	N_{rg}	#Opt	GAP	CPU	N_{rg}	#Opt
30	10	0.00	0.01	40.2	5	0.00	0.00	7.2	5
30	25	0.00	146.12	1386.4	5	0.00	0.13	71.2	5
30	50	25.96	3632.99	13670.2	0	0.00	1.04	174.2	5
50	10	3.08	1103	1526.2	4	0.00	0.77	180.4	5
50	25	98.93	3608.96	15263.4	0	0.00	374.79	4546	5
50	50	149.99	3598.28	17062.4	0	59.09	3732	1646	0
70	10	51.93	3634.55	9412.4	0	16.91	2694.41	10177.6	2
70	25	157.60	3621.21	10507.8	0	64.78	3725.01	1704.8	0
70	50	214.94	3619.20	11126.6	0	73.98	4024.32	291.4	0
100	10	93.45	3629.34	6742.6	0	57.03	3647.89	4440.4	0
100	25	208.03	3621.92	7350.4	0	75.73	3984.69	372.4	0
100	50	295.78	3618.87	7587.6	0	81.96	4034.36	86.6	0
Média		108.30	2804.09	8473.01	1.16	35.79	2184.95	1974.85	2.25

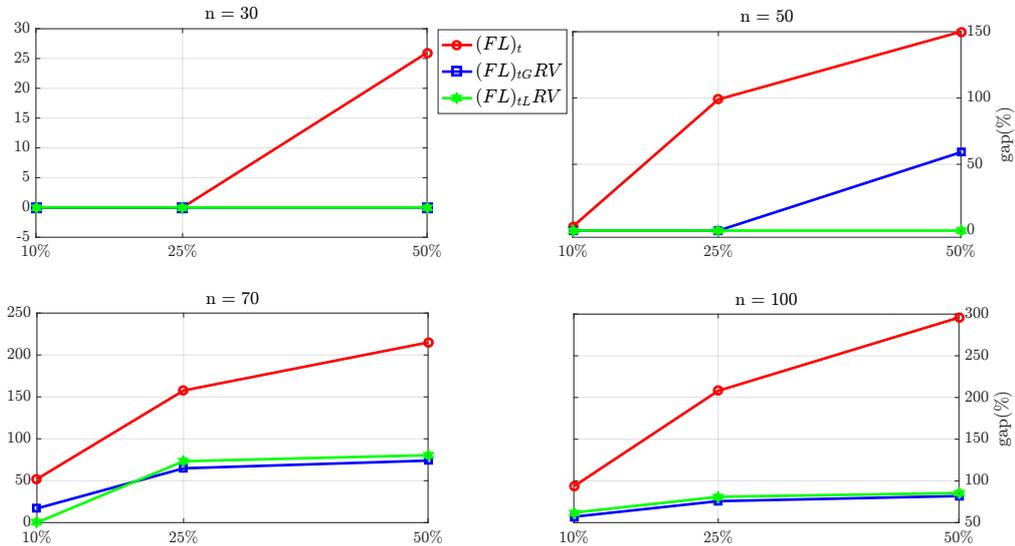
Fonte: Elaborado pelo autor.

Tabela 8: Comparação entre $(FL)_{tGRV}$ e $(FL)_{tLRV}$

n	d	$(FL)_{tGRV}$				$(FL)_{tLRV}$			
		GAP	CPU	N_{rg}	#Opt	GAP	CPU	N_{rg}	#Opt
30	10	0.00	0.00	7.2	5	0.00	0.00	11.8	5
30	25	0.00	0.13	71.2	5	0.00	0.09	234.4	5
30	50	0.00	1.04	174.2	5	0.00	0.70	284	5
50	10	0.00	0.77	180.4	5	0.00	0.88	3575.2	5
50	25	0.00	374.79	4546	5	0.00	25.62	14477.4	5
50	50	59.09	3732	1646	0	0.00	2355.82	33725	5
70	10	16.91	2694.41	10177.6	2	0.00	172.83	480601.4	5
70	25	64.78	3725.01	1704.8	0	73.23	3893.36	71146.8	0
70	50	73.98	4024.32	291.4	0	80.40	4018.58	29682	0
100	10	57.03	3647.89	4440.4	0	61.94	3732.44	486458.8	0
100	25	75.73	3984.69	372.4	0	80.92	3909.06	27130.2	0
100	50	81.96	4034.36	86.6	0	85.42	4067.38	25454.2	0
Média		35.79	2184.95	1974.85	2.25	31.82	1848.06	97731.77	2.91

Fonte: Elaborado pelo autor.

As figuras 11 e 12 mostram, respectivamente, a evolução dos gaps médios obtidos pelas formulações $(FL)_t$ com $(FL)_{tGRV}$, quando fixamos n e variamos a densidade e quando fixamos a densidade e variamos o número de vértices. Os gráficos das figuras 11 e 12 mostram que as desigualdades válidas ajudaram na diminuição do gap médio em todos os cenários avaliados, com maior diferença de porcentagem para a combinação $(n, d) = (100, 50)$.

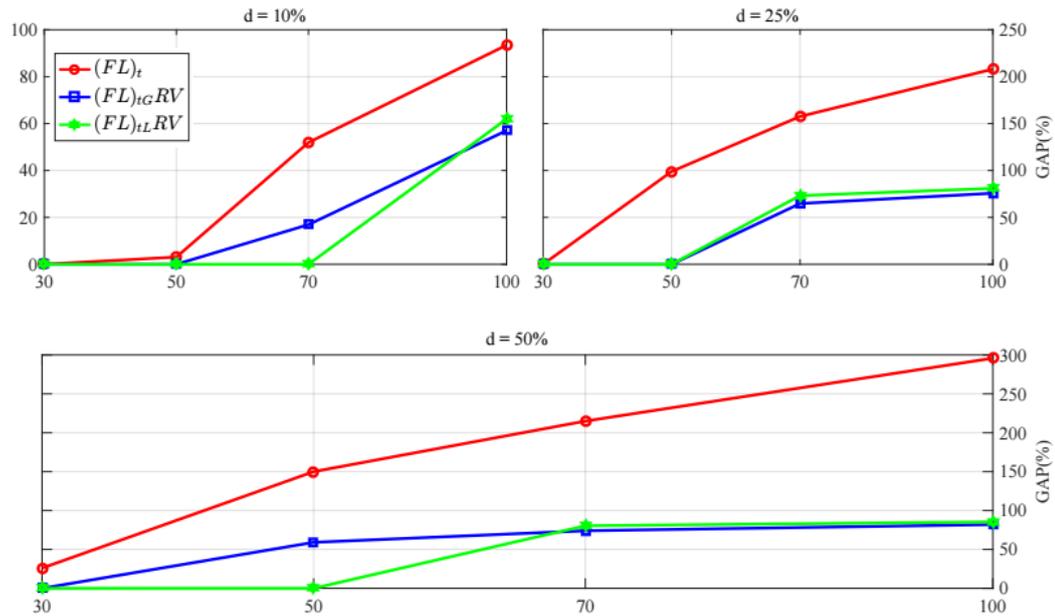
Figura 11: Gap médio com n fixo e variação de d 

Fonte: Elaborado pelo autor.

3.9 Conclusão

Apresentamos neste capítulo a teoria relacionada a dois métodos (t -linearização e suavização hiperbólica), aplicados pela primeira vez no contexto do problema do corte máximo.

Figura 12: Gap médio com d fixo e variação de n



Fonte: Elaborado pelo autor.

Em particular, para a aplicação da t -linearização precisamos estender, no Capítulo 2, resultados apresentados em Rodrigues (2010) e Rodrigues *et al.* (2012) para o problema quadrático da mochila. Essa extensão diz respeito à identificação das restrições t -linearizadas para substituir o termo quadrático, bem como de um procedimento polinomial para separá-las.

Revisitamos a teoria da suavização hiperbólica e, pela primeira vez, mostramos como aplicar essa técnica ao problema do corte máximo. A partir da formulação do Max-Cut com variáveis associadas aos vértices, obtivemos dois novos modelos suavizados, um deles incorporando as restrições triangulares, típicas da formulação com variáveis indexadas pelas arestas. Dessa maneira, combinamos características das duas formulações usuais.

Também derivamos aqui propriedades inéditas para as soluções ótimas de Max-Cut. Tais propriedades têm multiuso, podendo ser empregadas para fixação de variáveis, expressão de cortes lógicos, derivações de desigualdades válidas e critérios heurísticos.

Esses três elementos (t -linearização, suavização hiperbólica e propriedades específicas) foram usados separadamente ou combinados na proposição de algoritmos de solução, que foram submetidos a experimentos computacionais distintos.

Vimos no experimento 1 que o modelo linearizado pela t -linearização usa menos restrições e tempo relativamente maior de computação para obter o mesmo limite da linearização clássica e linearização Glover.

No experimento 2, vimos os desempenhos obtidos por três versões da suavização hiperbólica e duas heurísticas gulosas. Os resultados mostram melhores limites primais nos métodos que empregam busca local, assim como vantagem nos limites e tem-

pos computacionais, respectivamente, para a versão $SH + \Delta + BL$ e a heurística gulosa com critério de otimalidade.

Os resultados computacionais do experimento 3 comprovam que as restrições válidas propostas na Subseção 3.2.3 conseguiram melhorar significativamente o desempenho do modelo testado, para todas as combinações de vértices e densidade de arestas.

Vale destacar que diversas outras formas de uso dos ingredientes aqui apresentados podem ser pensadas. Algumas delas também foram avaliadas computacionalmente, embora os resultados não tenham sido aqui apresentados, porque se mostraram inferiores. Por exemplo, o uso das desigualdades válidas (38) e (39) não apenas na raiz mas também em cada nó da árvore de busca, atualizando os parâmetros M_k^0 e M_k^1 iterativamente, a partir de soluções dos modelos P_k^0 e P_k^1 .

Neste capítulo demos um passo a mais à cerca do conhecimento do problema do corte máximo. O valor de nosso trabalho reside na inovação do desenvolvimento teórico e na aplicação da t -linearização para custos arbitrários, na aplicação da suavização hiperbólica e no desenvolvimento de novas restrições, corroboradas pelos resultados práticos que mostraram a possibilidade de algoritmos competitivos, baseados na aplicação desses elementos.

Há várias direções de pesquisas futuras, relacionadas a esse contexto. Destacamos três delas:

1. Obtenção de restrições fortalecidas da t -linearização. A presença de restrições adicionais, além daquela criada pela transferência do termo quadrático da função objetivo, pode ser explorada para “apertar” as restrições da t -linearização. Podemos citar as próprias restrições aqui propostas, as desigualdades triangulares, ou mesmo uma restrição limitando a cardinalidade de uma das partes da bipartição;
2. Investigar a convergência da suavização hiperbólica com outros parâmetros. Aqui foi usada uma única valoração dos parâmetros γ , λ e τ para todos os termos da função objetivo e restrições. Como mencionado, podemos usar parâmetros diferentes para cada um deles. A calibração desses parâmetros demanda extensos e cuidadosos experimentos computacionais;
3. Exploração das restrições válidas. As restrições válidas tem potencialmente formas de uso mais variadas do que aquelas testadas nesse trabalho. Novamente, trata-se de um estudo computacional por si só relevante.

4 PROBLEMA DA DIVERSIDADE MÁXIMA

O problema da diversidade máxima – PDM pertence a uma família de problemas de dispersão e diversidade, cujo objetivo é identificar, a partir de um conjunto V com n elementos, um subconjunto S com m ($m < n$) elementos, de tal forma que a distância entre os pares dos elementos em S seja máxima (Zhou, Hao, e Duval, 2017). Genericamente, o PDM pode ser definido sobre um grafo não direcionado $G = (V, E)$, ponderado em arestas, onde $V = \{1, \dots, n\}$ e $c_{ij} \in \mathbb{R}_+$ denota o peso da aresta $\{i, j\}$. Por simplicidade, consideramos $c_{ij} = c_{ji}$ para $\{i, j\} \in E$, e $c_{ij} = 0$ quando $\{i, j\} \notin E$. Formalmente, PDM pode ser formulado como o seguinte problema quadrático binário (Kuo, Glover, e Dhir, 1993):

$$(F_1) \quad \max \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} x_i x_j \quad (63)$$

$$s.a \quad \sum_{i=1}^n x_i = m, \quad (64)$$

$$x_i \in \{0, 1\}, \quad 1 \leq i \leq n, \quad (65)$$

onde a variável binária $x_i = 1$ se o vértice $i \in S$ e $x_i = 0$ se $i \notin S$. A equação (64) garante que uma solução viável x contém exatamente m vértices.

Com o passar do tempo, PDM vem sendo amplamente estudado sobre diferentes nomenclaturas (Zhou, Hao, e Duval, 2017). Além disso, PDM também se revela um modelo útil para formular uma variedade de aplicações práticas. A Tabela 9 sumariza as principais aplicações desse problema.

Tabela 9 – Principais aplicações do problema da diversidade máxima

Contexto	Referência
Locação	Erkut e Neuman (1991)
Sistemas ecológicos	Pearce (1987)
Tratamentos médicos	Kuo, Glover, e Dhir (1993)
Genética	Porter <i>et al.</i> (1975)
Etnia	Swierenga (1977)
	McConnell (1988)
Design de Produto	(Ghyczy, 1985; Besanko, Perry, e Spady, 1990)

Fonte: Elaborado pelo autor.

Sendo um problema NP-Difícil (Kuo, Glover, e Dhir, 1993), há uma variedade de métodos de solução propostos na literatura, que podem ser divididos em duas principais categorias: algoritmos exatos e heurísticas. Na categoria dos algoritmos exatos, Erkut (1990) usou método branch-and-bound e conseguiu resolver instâncias com até 40 vértices. Já Pisinger (2006), fazendo uso de branch-and-bound com programação semidefinida,

conseguiu resolver problemas com até 90. Ainda na categoria dos exatos, Martí, Gallego, e Duarte (2010) propuseram um algoritmo brand-and-bound com ramificação n -ária, que não faz uso de modelo de programação matemática. Os autores também mostraram 4 propriedades para PDM, sendo 3 usadas pelo método na obtenção de bons limites superiores e a última usada como critério de poda. Com esse método, eles conseguiram resolver instâncias com 125 vértices para alguns valores de m . Além disso, usando o solver CPLEX 8.0 mostraram que as três formulações linear inteiras propostas por Kuo, Glover, e Dhir (1993) são capazes de resolver instâncias com até 30 vértices.

Para lidar com instâncias maiores, os algoritmos heurísticos são na maioria das vezes preferidos para encontrar soluções quase ótimas em um período de tempo aceitável. Os trabalhos iniciais sobre heurísticas para PDM foram baseados em abordagens gulosas construtivas e destrutivas (Aringhieri e Cordone, 2011). A partir daí, várias abordagens foram sendo aplicadas, com destaque para *Variable Neighborhood Search* – VNS, *Greedy Randomized Adaptive Search Procedure* – GRASP, *Iterative Tabu Search* – ITS, algoritmos evolucionários híbridos etc (Zhou, Hao, e Duval, 2017). Um *survey* sobre as heurísticas mais importantes anteriores a 2012 pode ser encontrado em Martí *et al.* (2013), onde os autores trabalharam com 30 métodos, 10 heurísticas e 20 metaheurísticas, comparando o tempo computacional de cada método no mesmo computador.

4.1 Formulações Lineares

Usando os procedimentos de linearização (clássica e Glover), descritos nas seções 2.1 e 2.2, Kuo, Glover, e Dhir (1993) propõem duas formulações lineares (F_2 e F_3) para PDM, como veremos abaixo:

$$(F_2) \quad \max \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} y_{ij} \quad (66)$$

$$s.a \quad \sum_{i=1}^n x_i = m, \quad (67)$$

$$y_{ij} \leq x_i, \quad 1 \leq i < j \leq n, \quad (68)$$

$$y_{ij} \leq x_j, \quad 1 \leq i < j \leq n, \quad (69)$$

$$x \in \{0, 1\}^n. \quad (70)$$

$$(F_3) \quad \max \sum_{i=1}^{n-1} u_i \quad (71)$$

$$s.a \quad u_i \leq \left(\sum_{j=i+1}^n c_{ij} \right) x_i, \quad 1 \leq i \leq n-1, \quad (72)$$

$$u_i \leq \sum_{j=i+1}^n c_{ij} x_j, \quad 1 \leq i \leq n-1, \quad (73)$$

$$\sum_{i=1}^n x_i = m, \quad (74)$$

$$x \in \{0, 1\}^n. \quad (75)$$

Como os custos são não negativos, note que, em (F_2) , as restrições $y_{ij} \geq x_i + x_j - 1$, para todo $1 \leq i < j \leq n$, são desnecessárias. Já na construção de (F_3) , observe que $\bar{U}_i = \sum_{j=i+1}^n c_{ij}$ e $\underline{U}_i = 0$, o que leva a tradução das restrições (5) e (6) em (72) e (73).

Em Macambira e Souza (2000), os autores usam uma modificação de (F_2) , reescrevendo a restrição de cardinalidade para obter:

$$(F_m) \quad \max \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} y_{ij} \quad (76)$$

$$s.a \quad \sum_{j \neq i} y_{ij} - (m-1)x_i = 0, \quad 1 \leq i \leq n, \quad (77)$$

$$y_{ij} \leq x_i, \quad 1 \leq i < j \leq n, \quad (78)$$

$$y_{ij} \leq x_j, \quad 1 \leq i < j \leq n, \quad (79)$$

$$x \in \{0, 1\}^n. \quad (80)$$

Podemos observar que, em relação a (F_1) , o modelo (F_2) acrescenta $n(n-1)/2$ novas variáveis e $n(n-1)$ restrições, enquanto que o modelo (F_3) acrescenta $n-1$ variáveis e $2(n-1)$ restrições. Já o modelo (F_m) , além de adicionar as mesmas variáveis e restrições que (F_2) , também substitui a restrição de cardinalidade (que só envolve x) por n restrições (que envolvem x e y).

4.2 Aplicação da t -Linearização ao PDM

Seguindo o desenvolvimento da Subseção 2.3.1, aplicamos a t -linearização ao PDM. Porém, neste caso fazemos uso da restrição de cardinalidade da solução para fortalecer as restrições da t -linearização.

Similarmente à Subseção 2.3.1, reescrevemos F_1 como $(F_1)_t$ da seguinte forma:

$$\begin{aligned} (F_1)_t \quad & \max \quad t \\ \text{s.a.} \quad & t \leq \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} x_i x_j, \\ & \sum_{i=1}^n x_i = m, \\ & (x, t) \in \mathbb{B}^n \times \mathbb{R}_+. \end{aligned}$$

Seja S_n o conjunto de todas as permutações de $\{1, \dots, n\}$. Note que

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} x_i x_j = \sum_{i=1}^n \sum_{j=i}^{i-1} c_{ji} x_i x_j = \sum_{i=1}^n \sum_{j=i}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(i)} x_{\pi(j)} \quad \forall \pi \in S_n.$$

Então, pelo Teorema 2.2, podemos reescrever $(F_1)_t$ como:

$$(F_1)_t^\pi \quad \max \quad t \tag{81}$$

$$\text{s.a.} \quad t \leq \sum_{i=1}^n \sum_{j=1}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(i)} \quad \forall \pi \in S_n, \tag{82}$$

$$\sum_{i=1}^n x_i = m, \tag{83}$$

$$(x, t) \in \mathbb{B}^n \times \mathbb{R}_+. \tag{84}$$

Mais ainda, a separação das restrições lineares que definem $(F_1)_t^\pi$ pode ser feita em tempo polinomial, como mostrado no Teorema 2.3.

4.3 Fortalecimento das Restrições t -Linearizadas

A presença da restrição de cardinalidade (83) permite fortalecer as restrições t -linearizadas. Sendo os pesos não negativos, elas podem ser substituídas conforme o seguinte teorema, que se baseia na ideia apresentada em Rodrigues *et al.* (2012) para o problema quadrático da mochila.

Teorema 4.1. *Para $i \in \{1, \dots, n\}$ e $\pi \in S_n$, seja $s_{\pi(i)}$ a soma dos $r := \min\{i-1, m-1\}$ maiores valores do conjunto $\{c_{\pi(j)\pi(i)} : j = 1, \dots, i-1\}$. Então as desigualdades*

$$t \leq \sum_{i=1}^n s_{\pi(i)} x_{\pi(i)}, \quad \text{para todo } \pi \in S_n, \tag{85}$$

são válidas para $(F_1)_t^\pi$.

Demonstração. Sejam $(x, t) \in \mathbb{B}^n \times \mathbb{R}_+$ viável para $(F_1)_t^\pi$ e $\pi \in S_n$. Então

$$\begin{aligned} t &\leq \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji} x_i x_j = \sum_{i=1}^n \sum_{j=1}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(i)} x_{\pi(j)} \\ &= \sum_{i=1}^n \left(\sum_{j=1}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(j)} \right) x_{\pi(i)} \end{aligned}$$

Como $x \in \mathbb{B}^n$ e $\sum_{i=1}^n x_{\pi(i)} = m$, se $x_{\pi(i)} = 1$ então $\sum_{j=1}^{i-1} x_{\pi(j)} \leq r$. Portanto se $x_{\pi(i)} = 1$, temos que $\sum_{j=1}^{i-1} c_{\pi(j)\pi(i)} x_{\pi(j)} \leq s_{\pi(i)}$. Segue então que $t \leq \sum_{i=1}^n s_{\pi(i)} x_{\pi(i)}$. \square

Observe que as restrições (85) dominam (82), pois $s_{\pi(i)} \leq \sum_{j=1}^{i-1} c_{\pi(j)\pi(i)}$.

4.4 Propriedades das Soluções Ótimas

Nesta seção, generalizamos resultados de Martí, Gallego, e Duarte (2010) sobre as soluções ótimas de PDM, que levam a critérios para fixação de variáveis e ensejam a expressões de restrições válidas.

4.4.1 Fixação de variáveis

Uma solução parcial \bar{x} para PDM é um vetor n -dimensional com algumas componentes fixadas em 0, até m componentes fixadas em 1 e as demais componentes livres. Dado uma solução parcial \bar{x} , definimos

$$\begin{aligned} V_0(\bar{x}) &= \{v \in V : \bar{x}_v = 0\} \\ V_1(\bar{x}) &= \{v \in V : \bar{x}_v = 1\} \\ V_2(\bar{x}) &= V \setminus (V_1(\bar{x}) \cup V_0(\bar{x})) \end{aligned}$$

Note que $|V_1(\bar{x})| \leq m$. Uma solução parcial \hat{x} é descendente de outra solução parcial \bar{x} se $V_1(\hat{x}) \supseteq V_1(\bar{x})$ e $V_0(\hat{x}) \supseteq V_0(\bar{x})$. A solução parcial é completa quando $|V_1(\bar{x})| = m$, $|V_0(\bar{x})| = n - m$ e $V_2(\bar{x}) = \emptyset$. Partindo de uma solução parcial \bar{x} , podemos obter uma solução completa descendente, transferindo $m - |V_1(\bar{x})|$ elementos de $V_2(\bar{x})$ para $V_1(\bar{x})$ e os demais para $V_0(\bar{x})$. Para melhor legibilidade de algumas expressões, usaremos nessa seção $c(a, b)$ para representar o peso da aresta entres os vértices a e b , ou seja, $c(a, b) = c_{ab}$.

Seja $S = \{s_1, s_2, \dots, s_m\} \subseteq V$ os vértices selecionados em uma solução completa x . Podemos expressar o valor a função objetivo (63) em x como

$$f(S) = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m c(s_i, s_j) = \sum_{j=1}^m c(s_j),$$

onde $c(s_i) = \frac{1}{2} \sum_{j=1}^m c(s_i, s_j)$ representa a contribuição do vértice s_i em $f(S)$. A partir dessa

expressão, podemos introduzir o potencial máximo ($c_{max}(v)$) e mínimo ($c_{min}(v)$) que um vértice v pode contribuir para qualquer solução, tais que:

$$c_{min}(v) = \frac{1}{2} \sum_{j=1}^{m-1} c(v, v_{\sigma(n-j)}), \quad c_{max}(v) = \frac{1}{2} \sum_{j=1}^{m-1} c(v, v_{\sigma(j)}),$$

e σ ordena os $n - 1$ vértices diferentes de v em ordem não crescente de pesos das arestas que os conectam a v . Observe que, a partir da definição acima, qualquer solução contendo o vértice v irá satisfazer as desigualdades

$$c_{min}(v) \leq c(v) \leq c_{max}(v), \quad (86)$$

mostradas e usadas por Martí, Gallego, e Duarte (2010) para estabelecer a Proposição 4.1 abaixo.

Proposição 4.1. *Dado uma solução ótima x e dois vértices $u, v \in V$, se $c_{max}(u) < c_{min}(v)$ e v não está selecionado na solução x , então u não pode ser selecionado em x .*

A Proposição 4.1 foi fortemente usada no desenvolvimento do algoritmo branch-and-bound proposto por Martí, Gallego, e Duarte (2010). Observe que podemos calcular melhores valores para c_{max} e c_{min} quando consideramos que alguns vértices já foram selecionados, ou seja, quando temos uma solução parcial \bar{x} .

Dada uma solução parcial \bar{x} e $v \in V_2(\bar{x})$, denotemos por $V_2^{max}(\bar{x}, v)$ e $V_2^{min}(\bar{x}, v)$, respectivamente, os $m - |V_1(\bar{x})| - 1$ vértices de $V_2(\bar{x}) \setminus \{v\}$ com os maiores e menores valores $c(v, w)$, para $w \in V_2(\bar{x}) \setminus \{v\}$. A partir dessas definições, vamos introduzir o potencial máximo ($c_{max}(\bar{x}, v)$) e mínimo ($c_{min}(\bar{x}, v)$) que um vértice $v \in V_2(\bar{x})$ pode contribuir, caso seja selecionado para fazer parte da nova solução parcial \hat{x} que descende de \bar{x} . Os potenciais são calculados pelas seguintes expressões:

$$c_{min}(\bar{x}, v) = \sum_{w \in V_1(\bar{x})} c(v, w) + \sum_{w \in V_2^{min}(\bar{x}, v)} c(v, w),$$

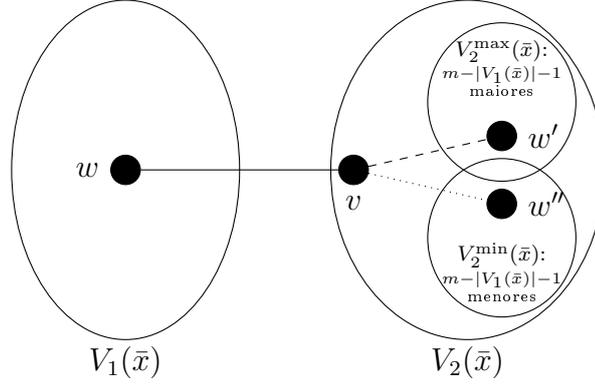
$$c_{max}(\bar{x}, v) = \sum_{w \in V_1(\bar{x})} c(v, w) + \sum_{w \in V_2^{max}(\bar{x}, v)} c(v, w).$$

A Figura 13 ilustra os termos dessas expressões. Com elas, derivamos propriedades de fixação de variáveis como a seguir.

Proposição 4.2. *Seja \bar{x} uma solução parcial e x^* uma melhor solução completa que descende de \bar{x} . Sejam $u, v \in V_2(\bar{x})$ tais que $c_{max}(\bar{x}, u) < c_{min}(\bar{x}, v)$. Se $x_v^* = 0$ então $x_u^* = 0$ (ou equivalentemente, se $x_u^* = 1$ então $x_v^* = 1$).*

Demonstração. Suponha, por absurdo, que $x_v^* = 0$ e $x_u^* = 1$. Então $V_1(x^*) = V_1(\bar{x}) \cup \{u\} \cup W$, onde $W \subseteq V_2(\bar{x}) \setminus \{u, v\}$, $|W| = m - |V_1(\bar{x})| - 1$, e $v \notin V_1(x^*)$. Defina \hat{x} uma solução completa que também descende de \bar{x} tal que $V_1(\hat{x}) = V_1(\bar{x}) \cup \{v\} \cup W$. Em outras palavras, \bar{x} troca u por v em x^* . Seja $f(x^*)$ o valor da solução completa x^* e $f(\hat{x})$ o valor

Figura 13: Ilustração das expressões de $c_{\min}(\bar{x}, v)$ e $c_{\max}(\bar{x}, v)$. Arestas pontilhadas compõem $c_{\min}(\bar{x}, v)$, arestas tracejadas compõem $c_{\max}(\bar{x}, v)$ e arestas cheias compõem ambos.



Fonte: Elaborado pelo autor.

da solução completa \hat{x} . Temos que

$$f(x^*) = \sum_{i \in V_1(\bar{x}) \cup W} \sum_{j \in V_1(\bar{x}) \cup W} c(i, j) + \sum_{w \in V_1(\bar{x})} c(u, w) + \sum_{w \in W} c(u, w),$$

$$f(\hat{x}) = \sum_{i \in V_1(\bar{x}) \cup W} \sum_{j \in V_1(\bar{x}) \cup W} c(i, j) + \sum_{w \in V_1(\bar{x})} c(v, w) + \sum_{w \in W} c(v, w),$$

ou ainda,

$$f(x^*) - f(\hat{x}) = \sum_{w \in V_1(\bar{x})} c(u, w) + \sum_{w \in W} c(u, w) - \left(\sum_{w \in V_1(\bar{x})} c(v, w) + \sum_{w \in W} c(v, w) \right).$$

Como

$$c_{\max}(\bar{x}, u) \geq \sum_{w \in V_1(\bar{x})} c(u, w) + \sum_{w \in W} c(u, w) \text{ e}$$

$$c_{\min}(\bar{x}, v) \leq \sum_{w \in V_1(\bar{x})} c(v, w) + \sum_{w \in W} c(v, w),$$

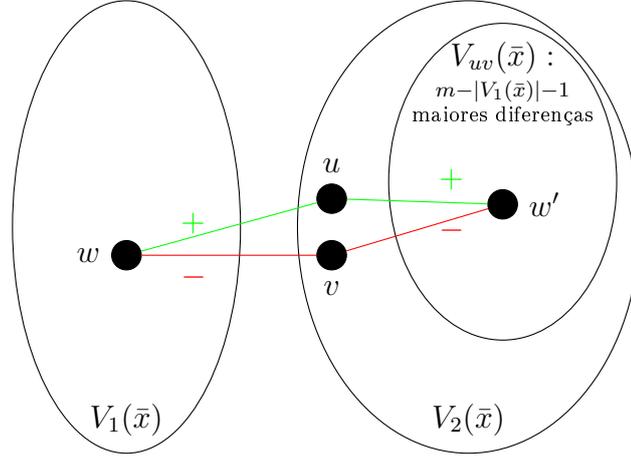
então $f(x^*) - f(\hat{x}) \leq c_{\max}(\bar{x}, u) - c_{\min}(\bar{x}, v) < 0$. Temos um absurdo, pois x^* é ótimo. \square

Podemos fortalecer a Proposição 4.2 como segue. Dados $u, v \in V_2(\bar{x})$, seja $V_{uv}(\bar{x})$ o conjunto com $m - |V_1(\bar{x})| - 1$ vértices $w \in V_2(\bar{x}) \setminus \{u, v\}$ com as maiores diferenças $[c(u, w) - c(v, w)]$ e defina

$$\delta_{uv}(\bar{x}) = \sum_{w \in V_1(\bar{x})} [c(u, w) - c(v, w)] + \sum_{w \in V_{uv}(\bar{x})} [c(u, w) - c(v, w)].$$

A Figura 14 ilustra os termos da expressão de $\delta_{uv}(\bar{x})$. Em particular, note que $\delta_{uv}(\bar{x}) \neq \delta_{vu}(\bar{x})$. A avaliação dessas expressões pode ser usada para fixação de variáveis, como segue.

Figura 14: Ilustração dos termos de $\delta_{uv}(\bar{x})$. Os sinais nas arestas indicam a contribuição de cada custo para a expressão.



Fonte: Elaborado pelo autor.

Proposição 4.3. *Seja \bar{x} uma solução parcial e x^* uma melhor solução completa que descende de \bar{x} . Sejam $u, v \in V_2(\bar{x})$ tais que $\delta_{uv}(\bar{x}) < 0$. Se $x_v^* = 0$ então $x_u^* = 0$ (ou equivalentemente Se $x_u^* = 1$ então $x_v^* = 1$).*

Demonstração. Suponha, por absurdo, que $x_v^* = 0$ e $x_u^* = 1$. Então $V_1(x^*) = V_1(\bar{x}) \cup \{u\} \cup W$, onde $W \subseteq V_2(\bar{x}) \setminus \{v, u\}$ e $|W| = m - |V_1(\bar{x})| - 1$. Note que $v \notin V_1(\bar{x})$. Defina a solução completa \hat{x} , que também descende de \bar{x} , tal que $V_1(\hat{x}) = V_1(\bar{x}) \cup \{v\} \cup \{W\}$. Temos que

$$\begin{aligned} f(x^*) - f(\hat{x}) &= \sum_{w \in V_1(\bar{x}) \cup W} c(u, w) - \sum_{w \in V_1(\bar{x}) \cup W} c(v, w) = \sum_{w \in V_1(\bar{x}) \cup W} [c(u, w) - c(v, w)] \\ &\leq \sum_{w \in V_1(\bar{x}) \cup V_{uv}(\bar{x})} [c(u, w) - c(v, w)] = \delta_{uv}(\bar{x}) < 0. \end{aligned}$$

Assim, chegamos a um absurdo, pois x^* é ótimo. \square

4.4.2 Restrições válidas

Alternativamente às proposições de fixação de variáveis, podemos usar os mesmos critérios para definir desigualdades válidas, como veremos na Proposição 4.4. Dadas uma solução parcial \bar{x} e $u, v \in V_2(\bar{x})$, definimos:

$$\begin{aligned} \dot{\delta}_{uv}(\bar{x}) &= \sum_{w \in V_1(\bar{x})} [c(u, w) - c(v, w)], \\ D_{uv}(\bar{x}) &= \{c(u, v) - c(v, w) : w \in V_2(\bar{x}) \setminus \{u, v\}\} \end{aligned}$$

e, para $k \in \{0, 1, 2\}$, $M_{uv}^k(\bar{x}) =$ soma dos $m - |V_1(\bar{x})| - k$ menores valores em $D_{uv}(\bar{x})$, se $|V_2(\bar{x}) \setminus \{u, v\}| \geq m - |V_1(\bar{x})| - k$, e $M_{uv}^k(\bar{x}) = \infty$, caso contrário. Seja ainda

$$M_{uv}(\bar{x}) = \min\{\dot{\delta}_{uv}(\bar{x}) + M_{uv}^2(\bar{x}), \dot{\delta}_{uv}(\bar{x})/2 + M_{uv}^1(\bar{x})/2, \dot{\delta}_{uv}(\bar{x}) + M_{uv}^0(\bar{x})\}.$$

Note que $M_{uv}^2(\bar{x}) = \infty \Rightarrow M_{uv}^1(\bar{x}) = \infty \Rightarrow M_{uv}^0(\bar{x}) = \infty$. Mais ainda M_{uv}^0 pode ser calculado a partir de M_{uv}^1 , que pode ser calculado a partir de M_{uv}^2 .

Proposição 4.4. *Sejam \bar{x} uma solução parcial e $u, v \in V_2(\bar{x})$. A desigualdade $\delta_{uv}(x, \bar{x}) \geq M_{uv}(\bar{x})(1 - x_u + x_v)$ é válida para as soluções completas de melhor valor que descendem de \bar{x} , onde*

$$\delta_{uv}(x, \bar{x}) = \sum_{w \in V_1(\bar{x})} [c(u, w) - c(v, w)] + \sum_{w \in V_2(\bar{x}) \setminus \{u, v\}} [c(u, w) - c(v, w)]x_w. \quad (87)$$

Demonstração. Seja x^* uma solução completa de melhor valor, descendente de \bar{x} . Seja $W = (V_1(x^*) \setminus V_1(\bar{x})) \setminus \{u, v\} \subseteq V_2(\bar{x}) \setminus \{u, v\}$. Observe que

$$\delta_{uv}(x^*, \bar{x}) = \dot{\delta}_{uv}(\bar{x}) + \sum_{w \in W} [c(u, w) - c(v, w)].$$

Considere os seguintes casos:

1. Se $x_u^* = 0, x_v^* = 0$: como $|W| = m - |V_1(\bar{x})|$, então $\delta_{uv}(x^*, \bar{x}) \geq \dot{\delta}_{uv}(\bar{x}) + M_{uv}^0(\bar{x}) \geq M_{uv}(\bar{x}) = M_{uv}(\bar{x})(1 - x_u^* + x_v^*)$;
2. Se $x_u^* = x_v^* = 1$: como $|W| = m - |V_1(\bar{x})| - 2$, então $\delta_{uv}(x^*, \bar{x}) \geq \dot{\delta}_{uv}(\bar{x}) + M_{uv}^2(\bar{x}) \geq M_{uv}(\bar{x}) = M_{uv}(\bar{x})(1 - x_u^* + x_v^*)$;
3. Se $x_u^* = 0, x_v^* = 1$: como $|W| = m - |V_1(\bar{x})| - 1$, então $\delta_{uv}(x^*, \bar{x}) \geq \dot{\delta}_{uv}(\bar{x}) + M_{uv}^1(\bar{x}) \geq 2M_{uv}(\bar{x}) = M_{uv}(\bar{x})(1 - x_u^* + x_v^*)$;
4. Se $x_u^* = 1, x_v^* = 0$: Obtenha solução completa \hat{x} , que também descende de \bar{x} , tal que $V_1(\hat{x}) = (V_1(x^*) \setminus \{u\}) \cup \{v\}$. Note que $V_1(x^*) \setminus \{u\} = V_1(\bar{x}) \cup \{w \in V_2(\bar{x}) \setminus \{u, v\} : x_w^* = 1\}$. Temos que $f(x) \geq f(\hat{x})$ e, portanto,

$$\begin{aligned} 0 \leq f(x^*) - f(\hat{x}) &= \sum_{w \in V_1(x^*) \setminus \{u\}} c(u, w) - \sum_{w \in V_1(\hat{x}) \setminus \{u\}} c(v, w) \\ &= \sum_{w \in V_1(\bar{x})} [c(u, w) - c(v, w)] + \sum_{w \in V_2(\bar{x}) \setminus \{u, v\}} [c(u, w) - c(v, w)]x_w^* \\ &= \delta_{uv}(x, \bar{x}). \end{aligned}$$

Logo, $\delta_{uv}(x^*, \bar{x}) \geq 0 = M_{uv}(\bar{x})(1 - x_u^* + x_v^*)$.

□

4.5 Heurística com Ordenação e Busca Local

A heurística que propomos depende apenas da análise dos custos das arestas. Ela é composta por duas fases. A primeira fase é um algoritmo que ordena o potencial de cada vértice em ordem não crescente e escolhe os m vértices de maior potencial para fazer parte de S . Na segunda fase, para cada par $(i, j) \in S \times V \setminus S$, compara-se a contribuição δ_{ik} com δ_{jk} e atualiza-se $S \leftarrow S - \{i\} \cup \{j\}$ caso o segundo tenha maior valor. Essas trocas são avaliadas, até que nenhum resultado melhore. Um pseudocódigo pode ser visto no Algoritmo 5.

Algoritmo 5: Heurística de ordenação e busca local

Entrada: Grafo $G = (V, E)$, m , função de pesos $c : E \rightarrow \mathbb{R}_+$

Saída: $S \subseteq V$, $|S| = m$

início

$\forall v \in V, pot[v] \leftarrow \sum_{\substack{u \in V \\ u \neq v}} c_{uv};$

ordene(pot , 'Decrescente'); /* Ordena os vértices em ordem não crescente segundo pot */

$S \leftarrow m$ primeiros vértices de acordo com a ordenação pot ;

repita

$melhorValor \leftarrow 0$;

para cada $i \in S$ **faça**

para cada $j \in V \setminus S$ **faça**

$\Delta_{jk} \leftarrow 0, \delta_{ik} \leftarrow 0$;

para cada $k \in S$ **faça**

se $k \neq i$ **então**

$\Delta_{jk} \leftarrow \Delta_{jk} + c_{jk}$;

$\delta_{ik} \leftarrow \delta_{ik} + c_{ik}$;

fim

fim

se $\Delta_{jk} > \delta_{ik}$ **então**

$S \leftarrow S - \{i\} \cup \{j\}$;

$melhorValor \leftarrow 1$;

fim

fim

fim

até $melhorValor = 0$;

fim

4.6 Experimentos Computacionais

Apresentamos nesta seção três experimentos computacionais. No experimento 1, vamos observar a qualidade do limite superior obtido pela t -linearização aplicada à formulação (F_1) , com as restrições originais (82) e com as restrições fortalecidas (85).

O limite da t -linearização será comparado com a relaxação das formulações (F_m) , (F_2) e (F_3) . Já no experimento 2, vamos observar a força das restrições (87). E por fim, no experimento 3 iremos avaliar um branch-and-bound que faz uso das restrições (87), da t -linearização com restrições fortalecidas (85), e da heurística proposta na seção anterior.

Todos os experimentos computacionais foram realizados em um processador Intel[®] Core[™] i5 – 4570 com 3.20 GHz, 16 GB RAM e sistema operacional Ubuntu 16.04 LTS. Utilizamos três conjuntos de instâncias, descritas a seguir, que foram introduzidas por Glover, Kuo, e Dhir (1998) e Silva e Ochi (2004) e estão disponíveis em www.opticom.es/mdp.

SOM-a(Silva) : Consiste de 50 instâncias, onde os pesos das arestas foram sorteados no intervalo $[0, 9]$ a partir de uma distribuição uniforme, sendo 5 instâncias para cada par (n, m) , onde $n \in \{25, 50, 100, 125, 150\}$ e $m = 0.1n, 0.3n$.

GKD-a(Glover) : Consiste de 75 instâncias, onde os pesos das arestas foram calculados através da distância euclidiana de pontos gerados de forma aleatória, sendo 5 instâncias para cada par (n, m) , onde $n \in \{10, 15, 30\}$ e $m = 0.2n, 0.3n, 0.4n, 0.6n$ e $0.8n$.

GKD-b(Glover2) : Consiste de 50 instâncias, onde os pesos das arestas foram calculados através da distância euclidiana de pontos gerados de forma aleatória, sendo 5 instâncias para cada par (n, m) , onde $n \in \{25, 50, 100, 125, 150\}$ e $m = 0.1n, 0.3n$.

4.6.1 Limite Superior

O primeiro experimento foi conduzido para testar a qualidade do limite superior LS obtido quando relaxamos a integralidade nas formulações F_2 , F_3 , F_m , $(F_1)_t^\pi$ (geração das restrições (82)) e $(F_1)_t^\pi + RF$ (geração das restrições (85)). Para esse experimento, usamos as instâncias denominadas SOM-a(Silva) e GKD-b(Glover2). Para cada uma das formulações relaxadas, calculamos o $GAP = \frac{LS-ML}{ML} \times 100$, onde ML é o valor da melhor solução, disponível em www.opticom.es, encontrada até o momento, o tempo de computação em segundos, CPU, a quantidade de restrições da formulação, R_f , para F_2 , F_3 e F_m , e a quantidade de restrições geradas, N_{rg} , na obtenção de LS para $(F_1)_t^\pi$ e $(F_1)_t^\pi + RF$. Para implementar as formulações lineares relaxadas, usamos o software comercial IBM/ILOG CPLEX 12.7 integrado com Ambiente de Desenvolvimento Integrado (IDE - Integrated Development Environment) *Code::Blocks* através da linguagem C++ no sistema operacional ubuntu 16.04 LTS.

As tabelas 10 e 11 resumizam os resultados obtidos por $(F_1)_t^\pi$ e $(F_1)_t^\pi + RF$, nas instâncias Silva e Glover2, respectivamente, onde cada linha, a partir da 3, apresenta a média de 5 instâncias para cada par (n, m) . Observa-se que em todas as combinações (n, m) o GAP diminuiu consideravelmente, quando consideramos as restrições fortalecidas, chegando a um fator médio de 3.5 e 5.5 vezes, respectivamente. Por outro lado, a quantidade de restrições geradas e o tempo de CPU aumentaram em todas as instâncias.

Tabela 10: GAP%, CPU(s) e N_{rg} médio para as relaxações $(F_1)_t^\pi$ e $(F_1)_t^\pi + RF$ nas instâncias Silva

<i>Silva</i>		$(F_1)_t^\pi$			$(F_1)_t^\pi + RF$		
n	m	GAP	CPU	N_{rg}	GAP	CPU	N_{rg}
25	2	1120.26	0.00	25	230.92	0.00	33.4
25	7	154.03	0.01	25	78.56	0.01	45.8
50	5	556.42	0.03	50.4	120.78	0.06	90.2
50	15	154.62	0.04	50	86.90	0.07	90.6
100	10	575.33	0.43	100.6	125.68	1.00	200
100	30	173.10	0.49	100	99.91	1.29	230.8
125	12	622.45	1.04	125	134.72	3.77	328.4
125	37	179.55	1.22	125	103.49	4.08	342
150	15	609.13	2.20	150.6	138.06	6.38	342.8
150	45	180.61	2.54	150	105.86	5.78	351.6
<i>Média</i>		360.46	0.66	75.13	102.07	1.87	171.3

Fonte: Elaborada pelo autor.

Tabela 11: GAP%, CPU(s) e N_{rg} médio para as relaxações $(F_1)_t^\pi$ e $(F_1)_t^\pi + RF$ nas instâncias Glover2

<i>Glover2</i>		$(F_1)_t^\pi$			$(F_1)_t^\pi + RF$		
n	m	GAP	CPU	N_{rg}	GAP	CPU	N_{rg}
25	2	1406.74	0.00	25	203.61	0.00	30.4
25	7	227.93	0.00	25	79.73	0.00	57.6
50	5	814.18	0.01	50	118.25	0.02	75
50	15	187.34	0.01	50	71.99	0.13	198.4
100	10	746.78	0.15	100.4	102.75	0.21	162.8
100	30	184.90	0.16	100	65.09	4.80	572.2
125	12	733.37	0.36	127.6	104.55	1.28	303.6
125	37	198.47	0.43	125	66.65	9.20	636.2
150	15	735.04	0.78	150.8	97.86	2.73	337.8
150	45	196.48	0.91	150	66.35	19.79	773.4
<i>Média</i>		543.12	0.28	90.38	97.68	3.82	314.74

Fonte: Elaborada pelo autor.

Já as tabelas 12 e 13 sumarizam os resultados obtidos pelas formulações F_2 , F_3 , F_m e $(F_1)_t^\pi + RF$ nas instâncias Silva e Glover2, respectivamente, onde cada linha, a partir da 3, apresenta a média de 5 instâncias para cada par (n, m) . Destacamos em negrito os melhores resultados. Para as instâncias Silva, nota-se que $(F_1)_t^\pi + RF$ obteve os melhores limites em todas as combinações (n, m) , com gap percentual médio mais que 3 vezes menor que aquele produzido por F_m . O desempenho foi ainda melhor quando comparado às relaxações F_2 e F_3 , que obtiveram limites semelhantes em todas as combinações (n, m) . Análise similar pode ser feita com relação às instâncias Glover2. Nesse caso, $(F_1)_t^\pi + RF$ conseguiu os melhores limites dentre todas as combinações (n, m) , exceto na combinação $(25, 7)$, onde foi superado por F_m . Com isso, o gap percentual médio de $(F_1)_t^\pi + RF$ foi cerca de 2.5 vezes menor que o de F_m . Novamente, os limites de F_2 e F_3 se mantiveram semelhantes, porém inferiores àqueles gerados por F_m .

Para uma melhor compreensão dos resultados, as figuras 15 e 16 mostram gráficos de barras que descrevem o gap médio obtido pelas cinco relaxações nas instâncias

Silva e Glover2, respectivamente.

Tabela 12: GAP%, CPU(s), R_f e N_{rg} médio para as relaxações F_2 , F_3 , F_m e $(F_1)_t^\pi + RF$ nas instâncias Silva

<i>Silva</i>		F_2		F_3			F_m			$(F_1)_t^\pi + RF$			
<i>n</i>	<i>m</i>	GAP	CPU	R_f	GAP	CPU	R_f	GAP	CPU	R_f	GAP	CPU	N_{rg}
25	2	1120.26	0.02	600	1128.23	0.00	48	508.33	0.00	625	230.92	0.00	33.4
25	7	154.03	0.02	600	156.18	0.00	48	98.45	0.01	625	78.56	0.01	45.8
50	5	581.03	0.39	2450	586.25	0.02	98	430.08	0.14	2500	120.78	0.06	90.2
50	15	158.30	0.41	2450	159.16	0.02	98	113.14	0.70	2500	86.90	0.07	90.6
100	10	582.80	2.37	9900	584.91	0.25	198	503.47	1.11	10000	125.68	1.00	200
100	30	173.10	2.68	9900	173.91	0.24	198	134.90	1.14	10000	99.91	1.29	230.8
125	12	633.49	3.68	15500	635.98	0.57	248	562.43	2.38	15625	134.72	3.77	328.4
125	37	179.55	4.32	15500	179.99	0.56	248	143.21	2.51	15625	103.49	4.08	342
150	15	611.53	7.16	22350	613.46	1.04	298	559.67	4.03	23000	138.06	6.38	342.8
150	45	180.61	9.46	22350	181.24	1.35	298	144.64	4.73	23000	105.86	5.78	351.6
<i>Média</i>		364.56	2.54	10160	366.61	0.34	178	319.83	1.68	10350	102.07	1.87	171.3

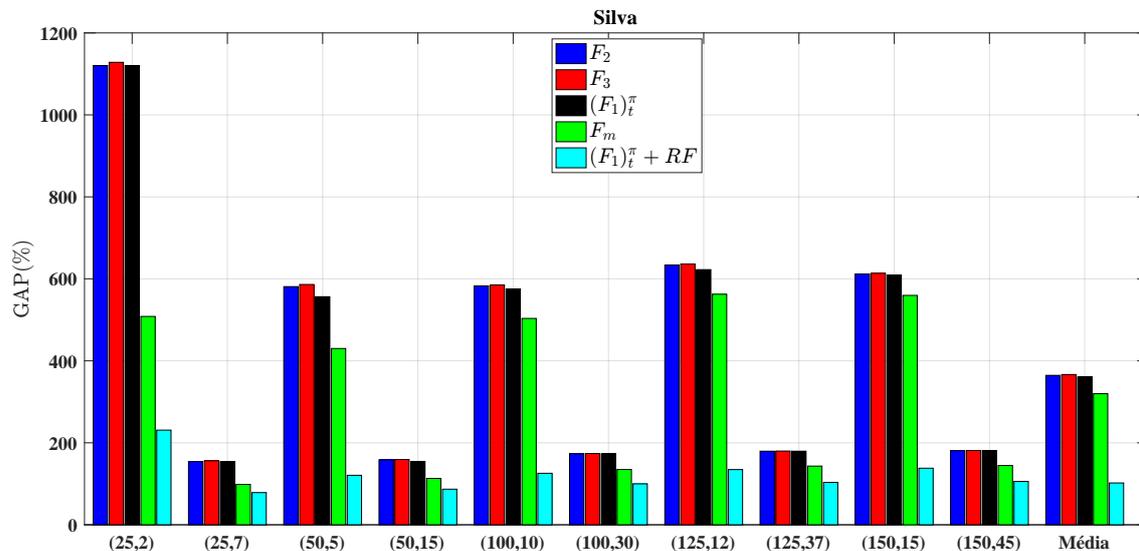
Fonte: Elaborada pelo autor.

Tabela 13: GAP%, CPU(s), R_f e N_{rg} médio para as relaxações F_2 , F_3 , F_m e $(F_1)_t^\pi + RF$ nas instâncias Glover2

<i>Glover2</i>		F_2		F_3			F_m			$(F_1)_t^\pi + RF$			
<i>n</i>	<i>m</i>	GAP	CPU	R_f	GAP	CPU	R_f	GAP	CPU	R_f	GAP	CPU	N_{rg}
25	2	1406.74	0.00	600	1411.98	0.00	48	386.61	0.00	625	203.61	0.00	30.4
25	7	227.93	0.01	600	228.54	0.00	48	74.73	0.02	625	79.73	0.00	57.6
50	5	814.18	0.05	2450	816.46	0.01	98	365.45	0.28	2500	118.25	0.02	75
50	15	187.34	0.05	2450	188.91	0.01	98	72.97	1.12	2500	71.99	0.13	198.4
100	10	746.78	1.26	9900	750.61	0.21	198	386.12	1.29	10000	102.75	0.21	162.8
100	30	184.90	1.38	9900	186.44	0.21	198	70.29	1.09	10000	65.09	4.80	572.2
125	12	733.37	1.63	15500	741.50	0.49	248	402.37	2.64	15625	104.55	1.28	303.6
125	37	198.47	1.91	15500	199.37	0.49	248	74.46	2.03	15625	66.65	9.20	636.2
150	15	735.04	2.99	22350	738.59	0.99	298	396.87	4.92	23000	97.86	2.73	337.8
150	45	196.48	3.28	22350	197.30	0.99	298	73.80	3.14	23000	66.35	19.79	773.4
<i>Média</i>		543.12	1.26	10160	545.97	0.34	178	230.37	1.65	10350	97.68	3.82	314.74

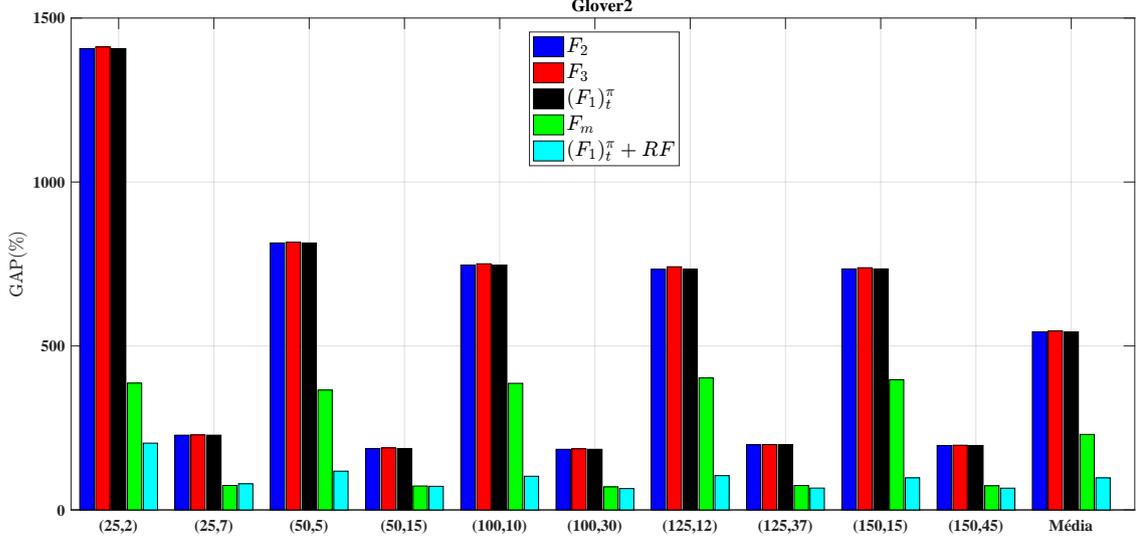
Fonte: Elaborada pelo autor.

Figura 15: Gap médio obtido pelas relaxações F_2 , F_3 , $(F_1)_t^\pi$, F_m e $(F_1)_t^\pi + RF$



Fonte: Elaborado pelo autor.

Figura 16: Gap médio obtido pelas relaxações F_2 , F_3 , $(F_1)_t^\pi$, F_m e $(F_1)_t^\pi + RF$



Fonte: Elaborado pelo autor.

4.6.2 Um algoritmo branch and bound

Nesta seção, consideramos dois experimentos com o método que iremos descrever a seguir.

O método exato aqui proposto para o PDM compõe-se de 4 procedimentos principais:

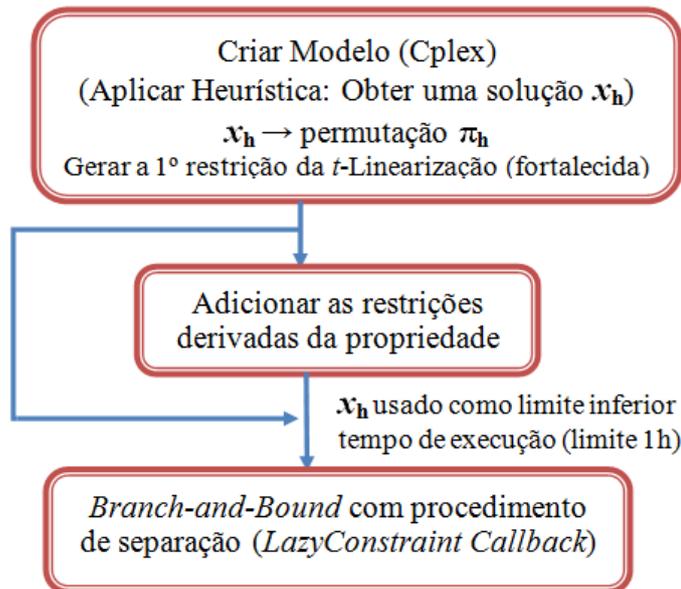
- 1– Conversão de (F_1) em $(F_1)_t^\pi$;
- 2– Separação das restrições de $(F_1)_t^\pi$ conforme Teorema 2.2, e fortalecimento da restrição violada segundo Teorema 4.1;
- 3– Obtenção de uma solução viável x_h (limite inferior) através da heurística da Seção 4.5, e aplicação do procedimento de separação em x_h , para gerar a primeira restrição fortalecida;
- 4– Adição à $(F_1)_t^\pi$ das desigualdades válidas $\delta_{uv}(x, \bar{x}) \geq M_{uv}(\bar{x})(1 - x_u + x_v)$, para $\bar{x} = \mathbf{0}$, obtendo assim o modelo $(BB)_t^\pi$:

$$\begin{aligned}
 (BB)_t^\pi \quad & \max \quad t \\
 \text{s.a} \quad & t \leq \sum_{i=1}^n s_{\pi(i)} x_{\pi(i)} \quad \forall \pi \in S_n, \\
 & \sum_{w \in V \setminus \{u, v\}} [c(u, w) - c(v, w)] x_w \geq M_{uv}(\mathbf{0})(1 - x_u + x_v) \quad \forall (u, v) \text{ e } \forall (v, u), \\
 & \sum_{i=1}^n x_i = m, \quad (x, t) \in \mathbb{B}^n \times \mathbb{R}_+.
 \end{aligned}$$

O primeiro experimento com o método exato foi conduzido para avaliar a força das desigualdades válidas (87). Para isso, testamos duas variações do nosso branch-and-bound. A primeira versão, denominada aqui de $(F_1)_t^\pi$, é composta apenas dos 3 primeiros procedimentos descritos no início desta subseção. A segunda versão, denominada aqui de $(BB)_t^\pi$, é composta por todos os 4 procedimentos.

É importante destacar que o procedimento de separação é usado dentro do branch-and-bound sempre que uma solução inteira é encontrada, ou seja, as restrições da t -linearização são usadas como *lazy constraints*. A Figura 17 mostra um esquema no formato de fluxograma do funcionamento do nosso método, apresentando as duas alternativas de fluxo, que originam $(F_1)_t^\pi$ ou $(BB)_t^\pi$.

Figura 17: Fluxograma do método exato. $(BB)_t^\pi$ segue as setas verticais. $(F_1)_t^\pi$ segue o caminho alternativo.



Fonte: Elaborado pelo autor.

Usamos o software comercial IBM/ILOG CPLEX 12.7 integrado com Ambiente de Desenvolvimento Integrado (IDE - Integrated Development Environment) *Code::Blocks* através da linguagem C++ para implementar as duas versões. Para essa comparação, utilizamos as instâncias denominadas de **GKD-a(Glover)** e limitamos o tempo computacional de execução de cada instância em 600 segundos.

Para cada instância calculamos o $GAP = \left[\frac{UB-LB}{LB} \times 100 \right]$, onde UB e LB são os limites inferiores e superiores respectivamente fornecidos pelo CPLEX. A Tabela 14 mostra, a partir da linha 3, a média do GAP, o tempo de computação em segundos, CPU, a quantidade de restrições geradas pelo procedimento de separação da t -linearização, N_{rg} , e a quantidade de soluções ótimas encontradas por cada formulação, #Opt. Destacamos em negrito os melhores resultados, conforme descrição dos experimentos.

Podemos observar que a formulação $(F_1)_t^\pi$ atingiu o tempo limite de execução, não conseguindo resolver de forma ótima as instâncias de combinação (30, 9), (30, 12) e (30, 18). Por outro lado, podemos notar que $(BB)_t^\pi$ conseguiu resolver de forma ótima todas as instâncias em curto período de tempo. A Figura 18 compara a quantidade média de restrições da t -linearização geradas por $(BB)_t^\pi$ e $(F_1)_t^\pi$. Podemos observar em $(BB)_t^\pi$ que o número de restrições t -linearizadas diminuiu consideravelmente em todas as

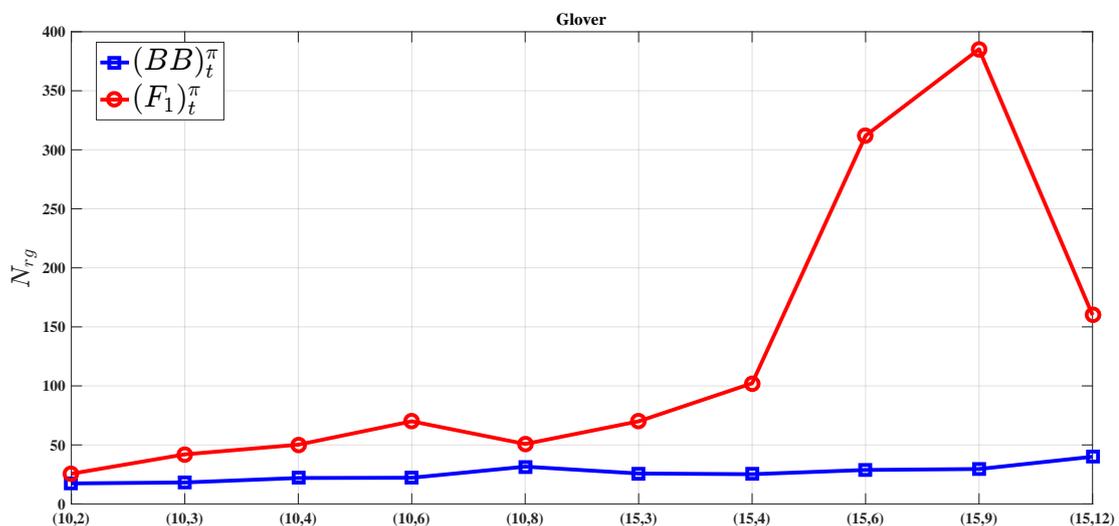
combinações de (n, m) . Para uma melhor visualização optamos por não colocar no gráfico da Figura 18 os resultados das combinações com $n = 30$.

Tabela 14 – Valores médios de GAP, CPU e N_{rg} e $\#Opt$ obtidos por $(F_1)_t^\pi$ e $(BB)_t^\pi$

<i>Glover</i>		$(F_1)_t^\pi$				$(BB)_t^\pi$			
n	m	GAP	CPU	N_{rg}	$\#Opt$	GAP	CPU	N_{rg}	$\#Opt$
10	2	0.00	0.005	25.6	5	0.00	0.007	17.4	5
10	3	0.00	0.011	42	5	0.00	0.009	18.2	5
10	4	0.00	0.011	50.2	5	0.00	0.010	22	5
10	6	0.00	0.016	70	5	0.00	0.010	22.2	5
10	8	0.00	0.010	50.8	5	0.00	0.010	31.6	5
15	3	0.00	0.018	70	5	0.00	0.020	25.8	5
15	4	0.00	0.032	102.2	5	0.00	0.024	25.2	5
15	6	0.00	0.143	312	5	0.00	0.024	28.8	5
15	9	0.00	0.175	385.2	5	0.00	0.021	29.6	5
15	12	0.00	0.040	160	5	0.00	0.020	40	5
30	6	0.00	8.661	1477.4	5	0.00	0.294	52.2	5
30	9	14.81	600	12551	0	0.00	0.463	55.8	5
30	12	18.81	600	13170	0	0.00	0.325	53.6	5
30	18	14.42	600	15101.6	0	0.00	0.279	76.4	5
30	24	0.00	254.34	8735.6	5	0.00	0.190	107	5
<i>Média</i>		3.20	137.55	3486.9	4	0.00	0.113	40.38	5

Fonte: Elaborada pelo autor.

Figura 18: Número médio de restrições geradas por $(F_1)_t^\pi$ e $(BB)_t^\pi$



Fonte: Elaborado pelo autor.

No segundo experimento, comparamos o desempenho do nosso *branch and bound* $(BB)_t^\pi$ com as formulações F_1 e F_m e com o *branch and bound* apresentado por Martí, Gallego, e Duarte (2010), denominado aqui por *BBmax*. Utilizamos a mesma máquina para melhor comparação entre os métodos: processador Intel[®] Core[™] i5-4570 com 3.20 GHz, 16 GB RAM e sistema operacional Ubuntu 16.04 LTS. Para implementar o $(BB)_t^\pi$, usamos o software comercial IBM/ILOG CPLEX 12.7, integrado com Ambiente

de Desenvolvimento Integrado (IDE - Integrated Development Environment) *Code::Blocks* e linguagem C++.

Algumas observações merecem ser feitas com respeito à implementação de *BBmax*. Infelizmente, os autores não mais dispunham do código original que pudesse ser executado agora. Sendo assim, a implementação aqui testada é nossa. A eficiência do *BBmax* é bastante influenciada pela escolha das estruturas de dados utilizadas para armazenar a árvore de enumeração e manter informações necessárias ao longo do processo. Observamos que *BBmax* não faz uso de um solver, como é o caso do $(BB)_t^\pi$, que emprega o software comercial IBM/ILOG CPLEX 12.7 para a resolução dos subproblemas gerados pela *t*-linearização.

Para essa comparação, utilizamos os conjuntos de instâncias denominados *Silva* e *Glover2*. Utilizamos dois parâmetros de medição: a prova de otimalidade Opt, ou seja, se o método conseguiu encontrar a solução ótima; e o tempo CPU em segundos, sendo esse último parâmetro limitado a 3600 segundos de execução em cada instância.

As tabelas 15 e 16 mostram, a partir da linha 3, o contraste entre os desempenhos das duas formulações e dos dois métodos (branch and bound) nas instâncias *Silva* e *Glover2*, respectivamente, onde destacamos em negrito os melhores resultados para cada instância. Para as instâncias *Silva*, o nosso método conseguiu provar a otimalidade em 32% (16 de 50) das instâncias testadas, enquanto que as formulações F_1 , F_m e o método *BBmax* o fizeram, respectivamente, em 30% (15 de 50), 50% (25 de 50) e 50% (25 de 50) dos casos. Podemos dizer que *BBmax* é o método mais indicado para as instâncias *Silva*, pois resolveu 50% delas utilizando em média menos tempo de execução.

Já para as instâncias *Glover2*, podemos ver que o nosso método conseguiu provar a otimalidade em 68% (34 de 50) das instâncias testadas, enquanto que as formulações F_1 , F_m e o método *BBmax* conseguiram, respectivamente, 30% (15 de 50), 50% (27 de 54) e 52% (26 de 50). É importante destacar também que os tempos de execução de $(BB)_t^\pi$ são melhores em quase todas as instâncias. Vale notar que alguns tempos de execução de F_1 e F_m são maiores que o limite de 3600s. Esses valores são explicados pelo fato de o CPLEX utilizar mais de uma *thread* de execução nas formulações. Tempos maiores que o limite também podem ser observados em $(BB)_t^\pi$, mas devido a outro motivo. Como utilizamos *lazycallback*, o CPLEX desabilita automaticamente o uso de mais de uma *thread* de execução. Nesse caso, o tempo extra se deve ao fato de o CPLEX contabilizar o tempo apenas a cada término de iteração.

Tabela 15: Resultados com as instâncias *Silva* obtidos pelas formulações F_1 e F_m , e os métodos $BBmax$ e $(BB)_t^\pi$

<i>Silva</i> (<i>n</i> , <i>m</i>)	F_1		F_m		$BBmax$		$(BB)_t^\pi$	
	CPU	Opt	CPU	Opt	CPU	Opt	CPU	Opt
(25, 2)	0.36	Sim	0.54	Sim	0.00	Sim	0.12	Sim
	0.48	Sim	0.49	Sim	0.00	Sim	0.12	Sim
	0.49	Sim	0.59	Sim	0.00	Sim	0.10	Sim
	0.36	Sim	0.63	Sim	0.00	Sim	0.16	Sim
	0.34	Sim	0.54	Sim	0.00	Sim	0.13	Sim
(25, 7)	2.56	Sim	0.74	Sim	0.02	Sim	0.28	Sim
	2.69	Sim	0.85	Sim	0.04	Sim	0.30	Sim
	3.63	Sim	0.68	Sim	0.03	Sim	0.46	Sim
	11.85	Sim	0.67	Sim	0.06	Sim	0.47	Sim
	3.70	Sim	0.37	Sim	0.01	Sim	0.27	Sim
(50, 5)	904.78	Sim	6.01	Sim	0.04	Sim	11.19	Sim
	927.06	Sim	5.46	Sim	0.04	Sim	7.88	Sim
	977.29	Sim	7.04	Sim	0.04	Sim	11.17	Sim
	1043.93	Sim	6.66	Sim	0.10	Sim	16.91	Sim
	860.28	Sim	5.56	Sim	0.04	Sim	12.99	Sim
(50, 15)	12687.39	Não	852.12	Sim	1878.82	Sim	3998.84	Não
	14160.92	Não	561.30	Sim	1259.63	Sim	3536.40	Sim
	13969.84	Não	934.00	Sim	2266.56	Sim	4007.68	Não
	12693.87	Não	811.69	Sim	2214.89	Sim	4027.77	Não
	13933.92	Não	486.83	Sim	1025.24	Sim	3998.16	Não
(100, 10)	13455.60	Não	7055.62	Sim	661.07	Sim	4250.67	Não
	13638.46	Não	4835.18	Sim	490.95	Sim	4198.82	Não
	13518.00	Não	4998.18	Sim	202.08	Sim	4244.96	Não
	13632.99	Não	4996.94	Sim	274.23	Sim	4235.89	Não
	13418.91	Não	8331.88	Sim	430.43	Sim	4262.54	Não
(100, 30)	14153.24	Não	13358.19	Não	3600	Não	4018.36	Não
	14090.52	Não	13420.96	Não	3600	Não	4001.35	Não
	14076.48	Não	13251.62	Não	3600	Não	3987.31	Não
	14107.39	Não	13323.43	Não	3600	Não	4022.90	Não
	14142.37	Não	13435.21	Não	3600	Não	4053.01	Não
(125, 12)	13453.20	Não	13973.61	Não	3600	Não	4350.71	Não
	14047.96	Não	13810.10	Não	3600	Não	4364.11	Não
	12415.78	Não	13890.66	Não	3600	Não	4358.19	Não
	14015.72	Não	13987.51	Não	3600	Não	4308.88	Não
	12249.24	Não	13904.22	Não	3600	Não	4380.29	Não
(125, 37)	14139.12	Não	13633.78	Não	3600	Não	4046.15	Não
	14170.82	Não	13777.70	Não	3600	Não	4027.24	Não
	14128.68	Não	13556.33	Não	3600	Não	4046.59	Não
	14130.82	Não	13667.29	Não	3600	Não	4073.58	Não
	14150.29	Não	13659.10	Não	3600	Não	4101.29	Não
(150, 15)	13547.16	Não	10081.66	Não	3600	Não	4287.87	Não
	13300.54	Não	9758.03	Não	3600	Não	3935.29	Não
	13755.22	Não	10326.32	Não	3600	Não	3959.62	Não
	13228.60	Não	10713.40	Não	3600	Não	4172.01	Não
	14093.98	Não	9618.89	Não	3600	Não	4269.75	Não
(150, 45)	14122.79	Não	7200.47	Não	3600	Não	4051.40	Não
	14094.53	Não	7200.53	Não	3600	Não	4022.86	Não
	14037.89	Não	7200.33	Não	3600	Não	3891.43	Não
	14125.20	Não	7200.37	Não	3600	Não	4034.21	Não
	14139.57	Não	7200.32	Não	3600	Não	3851.94	Não
<i>Média</i>	9715.33	15	6501.01	25	2014.07	25	2868.81	16

Fonte: Elaborada pelo autor.

Tabela 16: Resultados com as instâncias *Glover2* obtidos pelas formulações F_1 e F_m , e os métodos BB_{max} e $(BB)_t^\pi$

<i>Glover2</i> (<i>n</i> , <i>m</i>)	F_1		F_m		BB_{max}		$(BB)_t^\pi$	
	CPU	Opt	CPU	Opt	CPU	Opt	CPU	Opt
(25, 2)	0.46	Sim	0.56	Sim	0.00	Sim	0.06	Sim
	0.53	Sim	0.77	Sim	0.00	Sim	0.10	Sim
	0.45	Sim	0.50	Sim	0.00	Sim	0.06	Sim
	0.63	Sim	0.71	Sim	0.00	Sim	0.09	Sim
	0.49	Sim	0.60	Sim	0.00	Sim	0.07	Sim
(25, 7)	47.44	Sim	0.48	Sim	0.02	Sim	0.13	Sim
	52.99	Sim	0.58	Sim	0.02	Sim	0.16	Sim
	51.00	Sim	0.62	Sim	0.04	Sim	0.15	Sim
	8.94	Sim	0.54	Sim	0.02	Sim	0.12	Sim
	32.19	Sim	0.67	Sim	0.02	Sim	0.12	Sim
(50, 5)	2472.44	Sim	14.22	Sim	0.1	Sim	3.88	Sim
	2165.67	Sim	11.61	Sim	0.03	Sim	2.19	Sim
	1718.80	Sim	8.20	Sim	0.08	Sim	2.50	Sim
	1732.15	Sim	7.98	Sim	0.09	Sim	1.83	Sim
	2068.51	Sim	11.34	Sim	0.05	Sim	3.98	Sim
(50, 15)	12664.57	Não	62.91	Sim	304.85	Sim	4.90	Sim
	14193.77	Não	404.76	Sim	2304.34	Sim	3.13	Sim
	13250.65	Não	44.10	Sim	220.24	Sim	11.25	Sim
	13161.77	Não	45.62	Sim	369.90	Sim	8.92	Sim
	5361.46	Não	85.88	Sim	503.35	Sim	21.87	Sim
(100, 10)	13425.90	Não	1205.57	Sim	532.90	Sim	122.21	Sim
	13598.37	Não	1268.51	Sim	151.92	Sim	496.09	Sim
	13984.98	Não	3402.01	Sim	1487.71	Sim	236.46	Sim
	14064.13	Não	1610.61	Sim	469.02	Sim	840.03	Sim
	13834.26	Não	1248.18	Sim	246.63	Sim	483.41	Sim
(100, 30)	14174.95	Não	13647.26	Não	3600	Não	303.44	Sim
	13937.29	Não	13767.69	Não	3600	Não	515.81	Sim
	13690.92	Não	3012.10	Sim	3600	Não	252.39	Sim
	13382.39	Não	13757.56	Não	3600	Não	1074.80	Sim
	13748.66	Não	13839.38	Não	3600	Não	976.64	Sim
(125, 12)	14161.38	Não	13652.98	Não	3600	Não	127.17	Sim
	13634.60	Não	13610.17	Não	3600	Não	3769.68	Não
	14116.68	Não	13771.02	Não	3600	Não	895.98	Sim
	13920.12	Não	3198.23	Sim	749.30	Sim	1383.06	Sim
	14100.13	Não	13362.54	Não	3600	Não	1531.79	Sim
(125, 37)	14122.40	Não	14084.60	Não	3600	Não	3666.33	Não
	14049.88	Não	13539.52	Não	3600	Não	3603.65	Não
	14152.49	Não	13761.67	Não	3600	Não	3602.92	Não
	14025.75	Não	14098.41	Não	3600	Não	3610.59	Não
	14212.96	Não	13751.40	Não	3600	Não	3649.91	Não
(150, 15)	14076.63	Não	10042.73	Não	3600	Não	3676.78	Não
	13535.76	Não	10136.34	Não	3600	Não	3702.95	Não
	13881.07	Não	10275.89	Não	3600	Não	3681.58	Não
	13854.80	Não	9515.53	Não	3600	Não	3703.39	Não
	13826.41	Não	8773.59	Não	3600	Não	3704.89	Não
(150, 45)	14182.67	Não	7199.90	Não	3600	Não	3604.44	Não
	13214.28	Não	7200.46	Não	3600	Não	3604.56	Não
	14175.87	Não	10280.31	Não	3600	Não	3603.04	Não
	14050.27	Não	7200.86	Não	3600	Não	3603.09	Não
	14016.16	Não	7200.58	Não	3600	Não	3607.15	Não
<i>Média</i>	9722.74	15	5642.36	27	1874.81	26	1355.99	34

Fonte: Elaborada pelo autor.

4.7 Conclusão

Nesse capítulo, apresentamos contribuições para o estudo e resolução do problema da diversidade máxima (PDM). Mostramos como gerar restrições mais fortes na t -linearização e comprovamos, no experimento 1, que essas restrições fornecem, embora ainda alto, um

limite superior muito melhor quando comparado com outras formulações lineares da literatura. Derivamos também novas restrições para o PDM e comprovamos, no experimento 2, que, quando adicionadas à formulação da t -linearização, o tempo requerido para encontrar a solução ótima diminuiu consideravelmente, mostrando o potencial das novas restrições. Desenvolvemos ainda um método baseado em um branch-and-bound que calcula soluções exatas para o problema. O algoritmo proposto emprega os três elementos principais mencionados acima, juntamente com uma heurística.

Os experimentos computacionais indicam que nosso algoritmo é capaz de resolver, de forma ótima, instâncias de tamanho moderado(até 125 vértices). Comparamos nosso método com as formulações F_1 e F_m , e com uma implementação nossa do melhor método exato, que não utiliza solver, proposto na literatura, ou seja, o branch and bound desenvolvido por Martí, Gallego, e Duarte (2010). Os resultados favorecem o método aqui proposto para as instâncias do tipo Glover2. No entanto, podemos notar que nenhuma das formulações e dos métodos conseguiu resolver de forma ótima instâncias de tamanhos maiores. Vale destacar que nosso método usa fortemente o CPLEX, enquanto o BB_{max} não possui essa dependência e seu desempenho computacional é mais influenciado pela implementação adotada. Mesmo considerando que uma implementação mais cuidadosa possa reduzir o tempo computacional de BB_{max} , a diferença observada em nossos experimentos parece manter a vantagem para o novo método.

O valor de nosso trabalho reside na inovação dos três elementos propostos, corroborados pelo desenvolvimento teórico apresentado e pelos resultados práticos que se mostraram competitivos na aplicação desses elementos, o que possibilita futuras explorações nessa mesma direção. Vale destacar que os resultados teóricos da Seção 4.4 junto com parte dos experimentos computacionais compõem um trabalho aceito para publicação no Latin-Iberoamerican Conference on Operations Research CLAIO - 2018 (Soares, Campêlo, e Rebouças, 2018).

Há algumas direções de pesquisas futuras, relacionadas a esse contexto. Destacamos duas delas:

1. Efetuar o procedimento de separação nas restrições t -linearizadas já fortalecidas. Atualmente o procedimento de separação encontra a restrição mais violadas e em seguida é feito o fortalecimento;
2. Investigar o potencial do nosso método fazendo uso das restrições válidas à medida que variáveis forem sendo fixadas na árvore de enumeração. Como mencionado no experimento 2, as restrições válidas são adicionadas apenas na raiz, ou seja, nenhuma variável estava fixada nesse momento.

5 CLIQUE MÁXIMA PONDERADA EM ARESTA

O problema da m -clique ponderada máxima (Weighted Maximal m -Clique Problem – WCP) consiste em encontrar uma clique de maior peso com no máximo m vértices em um grafo completo ponderado nos vértices e nas arestas (Sørensen, 2004). Uma variação do WCP pode ser obtida quando apenas as arestas são ponderadas. Essa versão é também conhecida como problema da clique máxima ponderada em aresta – PCMPA (Dijkhuizen e Faigle, 1993), que formalmente pode ser modelado como um problema quadrático binário (Kuo, Glover, e Dhir, 1993; Alidaee *et al.*, 2007).

Dado um grafo completo não direcionado $G = (V, E)$, com $|V| = n$ vértices, ponderado em arestas, e um inteiro $m < n$, PCMPA consiste em encontrar uma clique em G , com no máximo m vértices, tal que a soma dos pesos das arestas seja máximo. Em outras palavras, deseja-se escolher um subconjunto $S \subseteq V$, com $|S| \leq m$, onde as arestas com ambas as extremidades em S somem o maior peso.

Diferentes versões/variações do PCMPA têm sido estudadas, originando diversas nomenclaturas, como mostra a Tabela 17. Elas se diferenciam pela restrição de cardinalidade (no máximo ou exatamente m vértices são desejados), pelo domínio dos pesos associados às arestas (reais, inteiros ou binários) e pela possibilidade do grafo ser ou não completo. Esse último ponto, porém, costuma ser tratado adicionando as arestas inexistentes com custos apropriados, de modo a gerar uma instância equivalente num grafo completo.

Tabela 17: Versões do PCMPA

Nomeclatura	$ S $	pesos	Referência
Maximum Edge-Weighted Clique	$\leq m$	$\in \mathbb{R}$	Macambira e Souza (2000)
Maximum Edge-Weighted Subgraph Heaviest Subgraph Problem	$= m$	$\in \mathbb{R}_+$	Macambira (2002)
Maximum Diversity Problem(MDP)			Billionnet (2005)
Maxsum, Dispersion			Martí, Gallego, e Duarte (2010)
Dense m -Subgraph	$= m$	$\in \mathbb{B}$	Ravi, Rosenkrantz, e Tayi (1994)
Maximum Edge Subgraph			Feige, Peleg, e Kortsarz (2001)
			Bonomo <i>et al.</i> (2009)

Fonte: Elaborada pelo autor.

A partir de um modelo quadrático binário, apresentamos cinco formulações para PCMPA, sendo uma linear inteira mista e o restante linear inteira. Todas podem ser facilmente adaptadas para as variações apresentadas na Tabela 17 ou mesmo para o caso onde pesos também são considerados nos vértices. As quatro primeiras formulações aqui mostradas são conhecidas da literatura, sendo três delas adaptações de modelos vistos no capítulo anterior; a quarta advém da modificação da restrição de cardinalidade e a última da aplicação da t -linearização ao modelo quadrático.

5.1 Formulações

Seja $V(G) = \{v_1, v_2, \dots, v_n\}$ e considere $c_e \in \mathbb{R}$ o peso de cada aresta $e \in E$. Por simplicidade, usamos indistintamente $c_{ji} = c_{ij}$ para representar o peso de $e = \{v_i, v_j\} \in E$. A formulação quadrática binária para PCMPA é dada naturalmente por:

$$(FCA_1) \quad \max \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} x_i x_j$$

$$\text{s.a.} \quad \sum_{i=1}^n x_i \leq m,$$

$$x_i \in \{0, 1\}, \quad 1 \leq i \leq n,$$

onde a variável binária $x_i = 1$ se o vértice $v_i \in S$ e $x_i = 0$ se $v_i \notin S$.

Usando procedimentos já vistos no capítulo anterior, (Kuo, Glover, e Dhir, 1993) propõem duas formulações lineares (F_2 e F_3) para o MDP, que também se aplicam ao PCMPA com pequenas adaptações. As versões adaptadas serão denominadas FCA_2 e FCA_3 , respectivamente.

A formulação (FCA_2) é obtida aplicando-se diretamente a linearização clássica a (FCA_1):

$$(FCA_2) \quad \max \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} y_{ij}$$

$$\text{s.a.} \quad \sum_{i=1}^n x_i \leq m,$$

$$y_{ij} \leq x_i, \quad 1 \leq i < j \leq n,$$

$$y_{ij} \leq x_j, \quad 1 \leq i < j \leq n,$$

$$y_{ij} \geq x_i + x_j - 1, \quad 1 \leq i < j \leq n,$$

$$x \in \{0, 1\}^n, \quad y \geq 0.$$

Em lugar de variáveis $y \in \mathbb{R}^{|E|}$, a formulação (FCA_3) usa uma variável $u \in \mathbb{R}^{n-1}$ tal que $u_i = x_i \sum_{j=i+1}^n c_{ij} x_j$. Com o intuito de obter tal igualdade, para todo $1 \leq i \leq n-1$, defina

$$\bar{U}_i = \sum_{j=i+1}^n \max(0, c_{ij}), \quad \underline{U}_i = \sum_{j=i+1}^n \min(0, c_{ij}).$$

Conforme desenvolvimento da Seção 2.2, obtemos então

$$\begin{aligned}
 (FCA_3) \quad & \max \sum_{i=1}^{n-1} u_i \\
 & \text{s.a.} \sum_{i=1}^n x_i \leq m, \\
 & u_i \leq \bar{U}_i x_i, \quad 1 \leq i \leq n-1, \\
 & u_i \leq \sum_{j=i+1}^n c_{ij} x_j - \underline{U}_i (1-x_i), \quad 1 \leq i \leq n-1, \\
 & x \in \{0, 1\}^n.
 \end{aligned}$$

Em Macambira e Souza (2000), os autores usam uma modificação de (FCA_2) , reescrevendo a restrição de cardinalidade para obter:

$$\begin{aligned}
 (FCA_m) \quad & \max \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} y_{ij} \\
 & \text{s.a.} \sum_{j \neq i} y_{ij} - (m-1)x_i \leq 0, \quad 1 \leq i \leq n, \\
 & y_{ij} \leq x_i, \quad y_{ij} \leq x_j, \quad y_{ij} \geq x_i + x_j - 1, \quad 1 \leq i < j \leq n, \\
 & x_i \in \{0, 1\}, \quad 1 \leq i \leq n, \quad y_{ij} \in \{0, 1\}, \quad 1 \leq i < j \leq n.
 \end{aligned}$$

Podemos observar que, em relação a FCA_1 , o modelo FCA_2 acrescenta $n(n-1)/2$ novas variáveis e $3n(n-1)/2$ restrições, enquanto que o modelo FCA_3 acrescenta $n-1$ variáveis e $2(n-1)$ restrições. Já o modelo FCA_m , além de adicionar as mesmas variáveis e restrições que FCA_2 , também substitui a restrição de cardinalidade (que só envolve x) por n restrições (que envolvem x e y).

5.2 Aplicação da t -Linearização ao PCMPA

Denote $a^+ = \max\{0, a\}$, $a^- = \min\{0, a\}$, $\forall a \in \mathbb{R}$, e $\bar{a} = 1 - a$, $\forall a \in [0, 1]$. Seja $F(x)$ a função objetivo de FCA_1 . Temos que

$$\begin{aligned}
 F(x) &= \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji} x_i x_j = \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji}^+ x_i x_j + \sum_{i=1}^n \sum_{j=1}^{i-1} \underbrace{c_{ji}^- x_i x_j}_{\frac{1}{2}c_{ji}^- x_i (1-\bar{x}_j) + \frac{1}{2}c_{ji}^- (1-\bar{x}_i) x_j} \\
 &= Q(x) + L(x)
 \end{aligned}$$

onde

$$Q(x) = \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji}^+ x_i x_j + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^{i-1} -c_{ji}^- (x_i \bar{x}_j + \bar{x}_i x_j),$$

$$L(x) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji}^- (x_i + x_j) = \sum_i \sum_{j<i} c_{ji}^- x_i + \sum_i \sum_{j>i} c_{ji}^- x_i = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n c_{ij}^- x_i.$$

Logo podemos reescrever FCA_1 como

$$(FCA_1)_t \quad \max \quad t + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n c_{ij}^- x_i$$

$$\text{s.a} \quad t \leq \sum_{i=1}^n \sum_{j=1}^{i-1} c_{ji}^+ x_i x_j + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^{i-1} -c_{ji}^- (x_i \bar{x}_j + \bar{x}_i x_j)$$

$$\sum_{i=1}^n x_i \leq m,$$

$$(x, t) \in \mathbb{B}^n \times \mathbb{R}, \bar{x}_j = 1 - x_j, \forall j \in \{1, \dots, n\}$$

e, pelo Teorema 2.5, podemos reescrever $(FCA_1)_t$ como

$$(FCA_1)_t^\pi \quad \max \quad t + \frac{1}{2} \sum_{i \in N} \sum_{j \in N} c_{ji}^- x_i \tag{88}$$

$$\text{s.a} \quad t \leq \frac{1}{2} \sum_{i \in N} \sum_{\substack{\pi(j) \in N_1 \\ j < i}} -c_{\pi(j)\pi(i)}^- (\bar{x}_{\pi(i)} + \bar{x}_{\pi(j)}) + \frac{1}{2} \sum_{i \in N} \sum_{\substack{\pi(j) \in N_0 \\ j < i}} -c_{\pi(j)\pi(i)}^- (x_{\pi(i)} + x_{\pi(j)})$$

$$+ \sum_{i \in N} \sum_{\substack{\pi(j) \in N_1 \\ j < i}} c_{\pi(j)\pi(i)}^+ x_{\pi(i)} + \sum_{i \in N} \sum_{\substack{\pi(j) \in N_0 \\ j < i}} c_{\pi(j)\pi(i)}^+ x_{\pi(j)} \quad \forall \pi \in S_n \quad \forall (N_0, N_1) \in \mathcal{N}, \tag{89}$$

$$\sum_{i=1}^n x_i \leq m, \tag{90}$$

$$(x, t) \in \mathbb{B}^n \times \mathbb{R}, \bar{x}_j = 1 - x_j, \forall j \in \{1, \dots, n\}, \tag{91}$$

onde S_n e \mathcal{N} é o conjunto das permutações e das partições de $\{1, \dots, n\}$, respectivamente.

Além disso, caso os custos sejam não negativos, o modelo pode ser simplificado para

$$\max t \quad \text{s.a:} \quad (90), (91), \quad t \leq \sum_i \sum_{j < i} c_{\pi(j)\pi(i)} x_{\pi(i)} \quad \forall \pi \in S_n, \tag{92}$$

como visto no capítulo anterior.

Distribuindo os termos da primeira parcela da desigualdade (89), podemos

reescrevê-la, com uma mudança na indexação das variáveis, como:

$$\sum_{i \in N} \sum_{\substack{\pi(j) \in N_1 \\ j < i}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- \bar{x}_{\pi(i)} + \sum_{\pi(i) \in N_1} \sum_{j > i} -\frac{1}{2} c_{\pi(j)\pi(i)}^- \bar{x}_{\pi(i)}.$$

Similarmente, reorganizando os termos da segunda, terceira e quarta parcelas em (89), obtemos:

$$\sum_{i \in N} \left(\sum_{\substack{\pi(j) \in N_0 \\ j < i}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- + \sum_{\substack{\pi(j) \in N_1 \\ j < i}} c_{\pi(j)\pi(i)}^+ \right) x_{\pi(i)} + \sum_{\pi(i) \in N_0} \sum_{j > i} \left(-\frac{1}{2} c_{\pi(j)\pi(i)}^- + c_{\pi(j)\pi(i)}^+ \right) x_{\pi(i)}.$$

Dessa forma, a restrição (89) evidenciando o coeficiente de cada variável e seu complemento é:

$$t \leq \sum_{\pi(i) \in N_0} \alpha_{\pi(i)} x_{\pi(i)} + \sum_{\pi(i) \in N_0} \bar{\alpha}_{\pi(i)} \bar{x}_{\pi(i)} + \sum_{\pi(i) \in N_1} \beta_{\pi(i)} x_{\pi(i)} + \sum_{\pi(i) \in N_1} \bar{\beta}_{\pi(i)} \bar{x}_{\pi(i)}, \quad (93)$$

onde

$$\alpha_{\pi(i)} = \sum_{\substack{\pi(j) \in N_0 \\ j < i}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- + \sum_{\substack{\pi(j) \in N_1 \\ j < i}} c_{\pi(j)\pi(i)}^+ + \sum_{j > i} \left(-\frac{1}{2} c_{\pi(j)\pi(i)}^- + c_{\pi(j)\pi(i)}^+ \right), \quad \bar{\alpha}_{\pi(i)} = \sum_{\substack{\pi(j) \in N_1 \\ j < i}} -\frac{1}{2} c_{\pi(j)\pi(i)}^-$$

$$\beta_{\pi(i)} = \sum_{\substack{\pi(j) \in N_0 \\ j < i}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- + \sum_{\substack{\pi(j) \in N_1 \\ j < i}} c_{\pi(j)\pi(i)}^+, \quad \bar{\beta}_{\pi(i)} = \sum_{\substack{\pi(j) \in N_1 \\ j < i}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- + \sum_{j > i} -\frac{1}{2} c_{\pi(j)\pi(i)}^-.$$

5.3 Fortalecimento das Restrições t -Linearizadas

A presença da restrição de cardinalidade (90) permite fortalecer as restrições t -linearizadas. Quando os custos são não negativos, elas podem ser substituídas conforme o Teorema 4.1, apresentado na Subseção 4.3, baseado na ideia apresentada em Rodrigues (2010) para o problema quadrático da mochila. Apresentamos a seguir uma generalização desse teorema para custos arbitrários. Esse resultado compõe um artigo publicado nos anais no Simpósio Brasileiro de Pesquisa Operacional - 2018 (Soares e Campêlo, 2018).

Teorema 5.1. *Seja $i \in \{1, \dots, n\}$, $\pi \in S_n$ e $(N_0, N_1) \in \mathcal{N}$. Defina*

$$S_{\pi(i)}^0 = \{c_{\pi(j)\pi(i)}^+ : (\pi(j) \in N_1, j < i) \text{ ou } j > i\}, \quad S_{\pi(i)}^1 = \{c_{\pi(j)\pi(i)}^+ : \pi(j) \in N_1, j < i\},$$

$$\bar{S}_{\pi(i)}^0 = \{-\frac{1}{2} c_{\pi(j)\pi(i)}^- : \pi(j) \in N_1, j < i\}, \quad \bar{S}_{\pi(i)}^1 = \{-\frac{1}{2} c_{\pi(j)\pi(i)}^- : (\pi(j) \in N_1, j < i) \text{ ou } j > i\}.$$

Para $k = 1, 2$, seja $s_{\pi(i)}^k$ a soma dos $\min\{m - 1, |S_{\pi(i)}^k|\}$ maiores valores de $S_{\pi(i)}^k$ e seja

$\bar{s}_{\pi(i)}^k$ a soma dos $\min\{m, |\bar{S}_{\pi(i)}^k|\}$ maiores valores de $\bar{S}_{\pi(i)}^k$. Então as desigualdades

$$\begin{aligned}
t \leq & \sum_{\pi(i) \in N_0} \left(s_{\pi(i)}^0 + \sum_{\substack{\pi(j) \in N_0 \\ j < i}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- + \sum_{j > i} -\frac{1}{2} c_{\pi(j)\pi(i)}^- \right) x_{\pi(i)} + \sum_{\pi(i) \in N_0} \bar{s}_{\pi(i)}^0 \bar{x}_{\pi(i)} \\
& + \sum_{\pi(i) \in N_1} \left(s_{\pi(i)}^1 + \sum_{\substack{\pi(j) \in N_0 \\ j < i}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- \right) x_{\pi(i)} + \sum_{\pi(i) \in N_1} \bar{s}_{\pi(i)}^1 \bar{x}_{\pi(i)}, \tag{94}
\end{aligned}$$

para todo $\pi \in S_n$ e todo $(N_0, N_1) \in \mathcal{N}$, são válidas para (88)-(91).

Demonstração. Sejam (x, t) viável para (88)-(91), $\pi \in S_n$ e $(N_0, N_1) \in \mathcal{N}$. Pelo Teorema 2.5, (x, t) satisfaz

$$\begin{aligned}
t \leq Q(x) &= \sum_i \sum_{j < i} c_{ji}^+ x_j x_i + \sum_i \sum_{j < i} -\frac{1}{2} c_{ji}^- (x_i \bar{x}_j + \bar{x}_i x_j) \\
&= \sum_i \sum_{j < i} c_{\pi(j)\pi(i)}^+ x_{\pi(j)} x_{\pi(i)} + \sum_i \sum_{j < i} -\frac{1}{2} c_{\pi(j)\pi(i)}^- (\bar{x}_{\pi(j)} x_{\pi(i)} + x_{\pi(j)} \bar{x}_{\pi(i)}).
\end{aligned}$$

Note que

$$\begin{aligned}
Q(x) &= \sum_i \sum_{\substack{j < i \\ \pi(j) \in N_1}} c_{\pi(j)\pi(i)}^+ x_{\pi(j)} x_{\pi(i)} + \sum_i \sum_{\substack{j < i \\ \pi(j) \in N_0}} c_{\pi(j)\pi(i)}^+ x_{\pi(j)} x_{\pi(i)} \\
&+ \sum_{\pi(i) \in N_0} \sum_{\substack{j < i \\ \pi(j) \in N_0}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- \bar{x}_{\pi(j)} x_{\pi(i)} + \sum_{\pi(i) \in N_1} \sum_{\substack{j < i \\ \pi(j) \in N_0}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- \bar{x}_{\pi(j)} x_{\pi(i)} \\
&+ \sum_{\pi(i) \in N_0} \sum_{\substack{j < i \\ \pi(j) \in N_1}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- \bar{x}_{\pi(j)} x_{\pi(i)} + \sum_{\pi(i) \in N_1} \sum_{\substack{j < i \\ \pi(j) \in N_1}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- x_{\pi(j)} \bar{x}_{\pi(i)} \\
&+ \sum_{\pi(i) \in N_1} \sum_{\substack{j < i \\ \pi(j) \in N_1}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- x_{\pi(j)} \bar{x}_{\pi(i)} + \sum_i \sum_{\substack{j < i \\ \pi(j) \in N_0}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- x_{\pi(j)} \bar{x}_{\pi(i)} \\
&= \sum_i \sum_{\substack{j < i \\ \pi(j) \in N_1}} c_{\pi(j)\pi(i)}^+ x_{\pi(j)} x_{\pi(i)} + \sum_{\pi(i) \in N_0} \sum_{j > i} c_{\pi(j)\pi(i)}^+ x_{\pi(j)} x_{\pi(i)} \\
&+ \sum_{\pi(i) \in N_0} \sum_{\substack{j < i \\ \pi(j) \in N_0}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- \bar{x}_{\pi(j)} x_{\pi(i)} + \sum_{\pi(i) \in N_1} \sum_{\substack{j < i \\ \pi(j) \in N_0}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- \bar{x}_{\pi(j)} x_{\pi(i)} \\
&+ \sum_{\pi(i) \in N_1} \sum_{j > i} -\frac{1}{2} c_{\pi(j)\pi(i)}^- x_{\pi(j)} \bar{x}_{\pi(i)} + \sum_{\pi(i) \in N_0} \sum_{\substack{j < i \\ \pi(j) \in N_1}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- x_{\pi(j)} \bar{x}_{\pi(i)} \\
&+ \sum_{\pi(i) \in N_1} \sum_{\substack{j < i \\ \pi(j) \in N_1}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- x_{\pi(j)} \bar{x}_{\pi(i)} + \sum_{\pi(i) \in N_0} \sum_{j > i} -\frac{1}{2} c_{\pi(j)\pi(i)}^- \bar{x}_{\pi(j)} x_{\pi(i)}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{\substack{\pi^{(i)} \in N_0 \\ x_{\pi^{(i)}} = 1}} \left(\sum_{\substack{\pi^{(j)} \in N_0 \\ j < i}} -\frac{1}{2} c_{\pi^{(j)\pi^{(i)}}}^- \bar{x}_{\pi^{(j)}} + \sum_{\substack{\pi^{(j)} \in N_1 \\ j < i}} c_{\pi^{(j)\pi^{(i)}}}^+ x_{\pi^{(j)}} + \sum_{j > i} -\frac{1}{2} c_{\pi^{(j)\pi^{(i)}}}^- \bar{x}_{\pi^{(j)}} + \sum_{j > i} c_{\pi^{(j)\pi^{(i)}}}^+ x_{\pi^{(j)}} \right) x_{\pi^{(i)}} \\
&+ \sum_{\substack{\pi^{(i)} \in N_0 \\ x_{\pi^{(i)}} = 0}} \left(\sum_{\substack{\pi^{(j)} \in N_1 \\ j < i}} -\frac{1}{2} c_{\pi^{(j)\pi^{(i)}}}^- x_{\pi^{(j)}} \right) \bar{x}_{\pi^{(i)}} + \sum_{\substack{\pi^{(i)} \in N_1 \\ x_{\pi^{(i)}} = 1}} \left(\sum_{\substack{\pi^{(j)} \in N_0 \\ j < i}} -\frac{1}{2} c_{\pi^{(j)\pi^{(i)}}}^- \bar{x}_{\pi^{(j)}} + \sum_{\substack{\pi^{(j)} \in N_1 \\ j < i}} c_{\pi^{(j)\pi^{(i)}}}^+ x_{\pi^{(j)}} \right) x_{\pi^{(i)}} \\
&+ \sum_{\substack{\pi^{(i)} \in N_1 \\ x_{\pi^{(i)}} = 0}} \left(\sum_{\substack{\pi^{(j)} \in N_1 \\ j < i}} -\frac{1}{2} c_{\pi^{(j)\pi^{(i)}}}^- x_{\pi^{(j)}} + \sum_{j > i} -\frac{1}{2} c_{\pi^{(j)\pi^{(i)}}}^- x_{\pi^{(j)}} \right) \bar{x}_{\pi^{(i)}}
\end{aligned}$$

Como $\sum_j x_{\pi^{(j)}} \leq m$, há no máximo $m - 1$ ou m variáveis $x_{\pi^{(j)}}$, $j \neq i$, iguais a 1, dependendo se $x_{\pi^{(i)}} = 1$ ou $x_{\pi^{(i)}} = 0$, respectivamente. Assim, calculando o máximo valor de cada termo entre parênteses na expressão de $Q(x)$ logo acima, obtemos os coeficientes da desigualdade (94). Observe que, em cada um desses termos, todos os coeficientes são não negativos e os conjuntos de índices dos somatórios são disjuntos (em particular, os dois últimos somatórios do primeiro termo são indexados por $j > i$, porém temos $c_{\pi^{(j)\pi^{(i)}}}^- = 0$ ou $c_{\pi^{(j)\pi^{(i)}}}^+ = 0$). Dessa forma, o máximo é obtido fazendo $\bar{x}_{\pi^{(j)}} = 1$ sempre, e um subconjunto de até m (se $x_{\pi^{(i)}} = 0$) ou $m - 1$ (se $x_{\pi^{(i)}} = 1$) variáveis $x_{\pi^{(j)}} = 1$, correspondendo às variáveis com os maiores coeficientes. \square

Observe que os coeficientes em (94) são menores ou iguais aos correspondentes em (93). Logo, as desigualdades (94) são um fortalecimento daquelas em (93).

5.4 Propriedades das Soluções Ótimas

Nesta seção, adaptamos e generalizamos para PCMPA os resultados obtidos para MDP na Seção 4.4 sobre soluções ótimas. Lembre que, para PCMPA, as soluções podem ter cardinalidade menor que m e os custos dos termos quadráticos são arbitrários. Evidentemente, quando esses custos são não negativos, podemos sempre nos restringir a soluções de cardinalidade m . Similarmente à Seção 4.4, as propriedades aqui derivadas levam a critérios para fixação de variáveis e à obtenção de desigualdades válidas.

5.4.1 Fixação de variáveis

Uma solução parcial para PCMPA é definida de forma idêntica àquela para MDP. Precisamente, é um vetor n -dimensional \bar{x} com algumas componentes fixadas em 0, até m componentes fixadas em 1 e as demais componentes livres. Novamente, dada uma solução parcial \bar{x} , definimos

$$\begin{aligned}
V_0(\bar{x}) &= \{v \in V : \bar{x}_v = 0\} \\
V_1(\bar{x}) &= \{v \in V : \bar{x}_v = 1\} \\
V_2(\bar{x}) &= V \setminus (V_1(\bar{x}) \cup V_0(\bar{x}))
\end{aligned}$$

Uma solução parcial \hat{x} é descendente de outra solução parcial \bar{x} se $V_1(\hat{x}) \supseteq V_1(\bar{x})$ e $V_0(\hat{x}) \supseteq V_0(\bar{x})$. Aqui, uma solução parcial é completa quando $V_2(\bar{x}) = \emptyset$. Para $\ell \leq m$, chamamos de ℓ -solução uma solução completa \bar{x} com $|V_1(\bar{x})| = \ell$. Partindo de uma solução parcial \bar{x} , podemos obter uma $|V_1(\bar{x})|$ -solução descendente, simplesmente transferindo os elementos de $V_2(\bar{x})$ para $V_0(\bar{x})$.

Como na Seção 4.4, $c(a, b)$ representa o peso da aresta entres os vértices a e b , ou seja, $c(a, b) = c_{ab}$. Além disso, para $d \in \mathbb{R}$, $d^+ = \max(0, d)$ e $d^- = \max(0, -d)$.

Dados um inteiro $\ell \leq m$, uma solução parcial \bar{x} e $v \in V_2(\bar{x})$, denotemos por $V_2^{\ell \max}(\bar{x}, v)$ e $V_2^{\ell \min}(\bar{x}, v)$, respectivamente, os $\ell - |V_1(\bar{x})| - 1$ vértices de $V_2(\bar{x}) \setminus \{v\}$ com os maiores e menores valores $c(v, w)$, para $w \in V_2(\bar{x}) \setminus \{v\}$. Os potenciais mínimo e máximo de contribuição de $v \in V_2(\bar{x})$, caso seja selecionado para fazer parte de uma ℓ -solução que descende de \bar{x} , são:

$$c_{min}^{\ell}(\bar{x}, v) = \sum_{w \in V_1(\bar{x})} c(v, w) + \sum_{w \in V_2^{\ell \min}(\bar{x}, v)} c(v, w),$$

$$c_{max}^{\ell}(\bar{x}, v) = \sum_{w \in V_1(\bar{x})} c(v, w) + \sum_{w \in V_2^{\ell \max}(\bar{x}, v)} c(v, w).$$

Adicionalmente, dado $u \in V_2(\bar{x}) \setminus \{v\}$, seja $V_{uv}^{\ell}(\bar{x})$ o conjunto com $\ell - |V_1(\bar{x})| - 1$ vértices $w \in V_2(\bar{x}) \setminus \{u, v\}$ com as maiores diferenças $[c(u, w) - c(v, w)]$ e defina

$$\delta_{uv}^{\ell}(\bar{x}) = \sum_{w \in V_1(\bar{x})} [c(u, w) - c(v, w)] + \sum_{w \in V_{uv}^{\ell}(\bar{x})} [c(u, w) - c(v, w)].$$

Em particular, note que $\delta_{uv}^{\ell}(\bar{x}) \neq \delta_{vu}^{\ell}(\bar{x})$.

Usando essas definições, derivamos os seguintes critérios para fixação de variáveis. As demonstrações serão omitidas porque são simples adaptações das provas das proposições 4.2 e 4.3.

Proposição 5.1. *Sejam \bar{x} uma solução parcial e x^* uma melhor ℓ -solução, $\ell \leq m$, que descende de \bar{x} . Sejam $u, v \in V_2(\bar{x})$ tais que $c_{max}^{\ell}(\bar{x}, u) < c_{min}^{\ell}(\bar{x}, v)$. Se $x_v^* = 0$ então $x_u^* = 0$ (ou equivalentemente, se $x_u^* = 1$ então $x_v^* = 1$).*

Proposição 5.2. *Sejam \bar{x} uma solução parcial e x^* uma melhor ℓ -solução, $\ell \leq m$, que descende de \bar{x} . Sejam $u, v \in V_2(\bar{x})$ tais que $\delta_{uv}^{\ell}(\bar{x}) < 0$. Se $x_v^* = 0$ então $x_u^* = 0$ (ou equivalentemente Se $x_u^* = 1$ então $x_v^* = 1$).*

Adicionalmente, derivamos critérios de fixação de variáveis para ℓ -soluções, válidos para todo valor de $\ell \leq m$. Para isso, definimos:

$$\delta_{uv}^{\oplus}(\bar{x}) = \sum_{w \in V_1(\bar{x})} [c(u, w) - c(v, w)] + \sum_{w \in V_{uv}^m(\bar{x})} [c(u, w) - c(v, w)]^+.$$

Observe que apenas as diferenças positivas são consideradas no segundo somatório e que

$$\delta_{uv}^\oplus(\bar{x}) \neq \delta_{vu}^\oplus(\bar{x}).$$

Proposição 5.3. *Sejam \bar{x} uma solução parcial e x^* uma melhor solução completa que descende de \bar{x} . Sejam $u, v \in V_2(\bar{x})$ tais que $\delta_{uv}^\oplus(\bar{x}) < 0$. Se $x_v^* = 0$ então $x_u^* = 0$ (ou equivalentemente Se $x_u^* = 1$ então $x_v^* = 1$).*

Demonstração. Suponha, por absurdo, que $x_v^* = 0$ e $x_u^* = 1$. Então $V_1(x^*) = V_1(\bar{x}) \cup \{u\} \cup W$, onde $W \subseteq V_2(\bar{x}) \setminus \{v, u\}$ e $|W| \leq m - |V_1(\bar{x})| - 1$. Note que $v \notin V_1(\bar{x})$. Defina a solução completa \hat{x} , que também descende de \bar{x} , tal que $V_1(\hat{x}) = V_1(\bar{x}) \cup \{v\} \cup \{W\}$. Temos que

$$\begin{aligned} f(x^*) - f(\hat{x}) &= \sum_{w \in V_1(\bar{x}) \cup W} c(u, w) - \sum_{w \in V_1(\bar{x}) \cup W} c(v, w) \\ &\leq \sum_{w \in V_1(\bar{x})} [c(u, w) - c(v, w)] + \sum_{w \in W} [c(u, w) - c(v, w)]^+ \\ &\leq \sum_{w \in V_1(\bar{x})} [c(u, w) - c(v, w)] + \sum_{w \in V_{uv}^m(\bar{x})} [c(u, w) - c(v, w)]^+ \\ &= \delta_{uv}^\oplus(\bar{x}) < 0. \end{aligned}$$

Temos um absurdo, pois x^* é ótimo. □

5.4.2 Restrições válidas

Similarmente à Subseção 4.4.2, derivamos agora desigualdades válidas para ℓ -soluções ótimas de PCMPA. Dadas uma solução parcial \bar{x} e $u, v \in V_2(\bar{x})$, como antes definimos:

$$\begin{aligned} \dot{\delta}_{uv}(\bar{x}) &= \sum_{w \in V_1(\bar{x})} [c(u, w) - c(v, w)] \quad \text{e} \\ D_{uv}(\bar{x}) &= \{c(u, w) - c(v, w) : w \in V_2(\bar{x}) \setminus \{u, v\}\}. \end{aligned}$$

Além disso, para $k \in \{0, 1, 2\}$ e $0 \leq \ell \leq m$, seja $M_{uv}^{\ell, k}(\bar{x})$ a soma dos $\ell - |V_1(\bar{x})| - k$ menores valores em $D_{uv}(\bar{x})$, se $|V_2(\bar{x}) \setminus \{u, v\}| \geq \ell - |V_1(\bar{x})| - k$, e $M_{uv}^{\ell, k}(\bar{x}) = \infty$, caso contrário. Seja ainda

$$M_{uv}^\ell(\bar{x}) = \min\{\dot{\delta}_{uv}(\bar{x}) + M_{uv}^{\ell, 2}(\bar{x}), \dot{\delta}_{uv}(\bar{x})/2 + M_{uv}^{\ell, 1}(\bar{x})/2, \dot{\delta}_{uv}(\bar{x}) + M_{uv}^{\ell, 0}(\bar{x})\}.$$

A generalização da Proposição 4.4, cuja prova segue os mesmos argumentos e, por isso, será omitida, é dada abaixo.

Proposição 5.4. *Sejam $\ell \leq m$, \bar{x} uma solução parcial e $u, v \in V_2(\bar{x})$. A desigualdade $\delta_{uv}(x, \bar{x}) \geq M_{uv}^\ell(\bar{x})(1 - x_u + x_v)$ é válida para as ℓ -soluções de melhor valor que descendem de \bar{x} , onde*

$$\delta_{uv}(x, \bar{x}) = \sum_{w \in V_1(\bar{x})} [c(u, w) - c(v, w)] + \sum_{w \in V_2(\bar{x}) \setminus \{u, v\}} [c(u, w) - c(v, w)]x_w. \quad (95)$$

Adicionalmente, temos as seguintes desigualdades válidas para ℓ -soluções, para qualquer $\ell \leq m$.

Proposição 5.5. *Sejam \bar{x} uma solução parcial e $u, v \in V_2(\bar{x})$. A desigualdade $\delta_{uv}^\oplus(x, \bar{x}) \geq \min(\dot{\delta}_{uv}(\bar{x}), \dot{\delta}_{uv}(\bar{x})/2)(1 - x_u + x_v)$ é válida para as soluções completas de melhor valor que descendem de \bar{x} , onde*

$$\delta_{uv}^\oplus(x, \bar{x}) = \sum_{w \in V_1(\bar{x})} [c(u, w) - c(v, w)] + \sum_{w \in V_2(\bar{x}) \setminus \{u, v\}} [c(u, w) - c(v, w)]^+ x_w. \quad (96)$$

Demonstração. Seja x uma solução completa de melhor valor, descendente de \bar{x} . Seja $W = (V_1(x) \setminus V_1(\bar{x})) \setminus \{u, v\} \subseteq V_2(\bar{x}) \setminus \{u, v\}$. Observe que

$$\delta_{uv}^\oplus(x, \bar{x}) = \dot{\delta}_{uv}(\bar{x}) + \sum_{w \in W} [c(u, w) - c(v, w)]^+ \geq \dot{\delta}_{uv}(\bar{x}).$$

A validade da desigualdade pode ser verificada, considerando os seguintes casos:

1. Se $x_u = x_v$: $\delta_{uv}^\oplus(x, \bar{x}) \geq \dot{\delta}_{uv}(\bar{x}) \geq \min(\dot{\delta}_{uv}(\bar{x}), \dot{\delta}_{uv}(\bar{x})/2)(1 - x_u + x_v)$;
2. Se $x_u = 0, x_v = 1$: $\delta_{uv}^\oplus(x, \bar{x}) \geq \dot{\delta}_{uv}(\bar{x}) = \frac{\dot{\delta}_{uv}(\bar{x})}{2}(1 - x_u + x_v) \geq \min(\dot{\delta}_{uv}(\bar{x}), \dot{\delta}_{uv}(\bar{x})/2)(1 - x_u + x_v)$;
3. Se $x_u = 1, x_v = 0$: Obtenha solução completa \hat{x} , que também descende de \bar{x} , tal que $V_1(\hat{x}) = (V_1(x) \setminus \{u\}) \cup \{v\}$. Note que $V_1(x) \setminus \{u\} = V_1(\bar{x}) \cup \{w \in V_2(\bar{x}) \setminus \{u, v\} : x_w = 1\}$. Temos que $f(x) \geq f(\hat{x})$ e, portanto,

$$\begin{aligned} 0 \leq f(x) - f(\hat{x}) &= \sum_{w \in V_1(x) \setminus \{u\}} c(u, w) - \sum_{w \in V_1(\hat{x}) \setminus \{u\}} c(v, w) \\ &= \sum_{w \in V_1(\bar{x})} [c(u, w) - c(v, w)] + \sum_{w \in V_2(\bar{x}) \setminus \{u, v\}} [c(u, w) - c(v, w)] x_w \\ &\leq \sum_{w \in V_1(\bar{x})} [c(u, w) - c(v, w)] + \sum_{w \in V_2(\bar{x}) \setminus \{u, v\}} [c(u, w) - c(v, w)]^+ x_w \\ &= \delta_{uv}^\oplus(x, \bar{x}). \end{aligned}$$

$$\text{Logo, } \delta_{uv}^\oplus(x, \bar{x}) \geq 0 = \min(\dot{\delta}_{uv}(\bar{x}), \dot{\delta}_{uv}(\bar{x})/2)(1 - x_u + x_v).$$

□

5.5 Heurística Gulosa

A heurística que propomos depende apenas de uma análise gulosa dos custos das arestas. Inicialmente escolhemos a aresta de maior custo e adicionamos as extremidades ao conjunto S . Em seguida, iniciamos um processo iterativo para escolher, dentre os vértices que não estão em S , aquele que mais tem a contribuir positivamente caso seja adicionado ao conjunto S . Esse processo é repetido enquanto existirem vértices com contribuição positiva, desde que $|S| \leq m$. Um pseudocódigo pode ser visto no Algoritmo 6.

Algoritmo 6: Heurística gulosa**Entrada:** Grafo $G = (V, E)$, $m \geq 2$, função de pesos $c : E \rightarrow \mathbb{R}$ **Saída:** $S \subseteq V$, $|S| \leq m$ **início** $c_{i^*j^*} = \max\{c_{ij} : i, j \in \{1, \dots, n\}\}$, $S \leftarrow \{i^*, j^*\}$, e $V' \leftarrow V \setminus \{i^*, j^*\}$; $melhorValor \leftarrow 1$;**enquanto** $melhorValor = 1$ e $|S| < m$ e $V' \neq \emptyset$ **faça** $melhorValor \leftarrow 0$, $maiorPotencial \leftarrow 0$;**para cada** $k \in V' \setminus S$ **faça** $\Delta_k \leftarrow \sum_{i \in S} c_{ki}$;**se** $\Delta_k > maiorPotencial$ **então** $maiorPotencial \leftarrow \Delta_k$; $v \leftarrow k$;**fim****fim****se** $maiorPotencial > 0$ **então** $S \leftarrow S \cup \{v\}$, $V' \leftarrow V' \setminus \{v\}$; $melhorValor \leftarrow 1$;**fim****fim****fim****return** S ;**5.6 Experimentos Computacionais**

Apresentamos nesta seção dois experimentos computacionais. No experimento 1, vamos observar a qualidade do limite superior obtido pela t -linearização aplicada a formulação (FCA_1) . O limite da t -linearização será comparado com a relaxação das formulações (FCA_2) , (FCA_3) e (FCA_m) . No experimento 2 iremos avaliar um branch-and-bound que faz uso das restrições (87), da t -linearização com restrições fortalecidas (94), e da heurística proposta na Subseção 5.5.

Para isso, utilizamos 60 instâncias geradas por Macambira e Souza (2000), a partir dos parâmetros $k \in \{1, \dots, 5\}$, $r \in (0, 1]$, $n \in \{40, 42, 44, 45, 46, 48\}$ e $m = \lfloor n/2 \rfloor$. Elas estão divididas em dois grupos. Cada uma das 30 instâncias do primeiro grupo é gerada a partir de uma das combinações (n, k) , e um valor aleatório de r , sendo os pesos das arestas positivos, gerados aleatoriamente no intervalo $1 \leq c_{ij} \leq 1000r^k$ ($1 \leq c_{ij} \leq 1000$). Similarmente, o segundo conjunto compreende 30 instâncias, cada uma gerada a partir de uma combinação (n, k) , e um valor aleatório r , onde os pesos das arestas podem ser positivos ou negativos, gerados aleatoriamente no intervalo $-1000r^k \leq c_{ij} \leq 1000r^k$ ($-1000 \leq c_{ij} \leq 1000$). As instâncias do primeiro grupo serão chamadas positivas e as do segundo grupo, mistas.

5.6.1 Limite superior

Os resultados do primeiro experimento serão apresentados nesta subseção. Ele visa avaliar a qualidade do limite superior LS , obtido quando relaxamos a integralidade das formulações FCA_2 , FCA_3 , FCA_m e $(FCA_1)_t^\pi$.

Calculamos para cada uma das formulações relaxadas o $GAP = (\frac{LS-OPT}{OPT} \times 100)$, onde LS é o valor da relaxação e OPT é valor da solução ótima para o problema, disponível em Macambira e Souza (2000). Determinamos também a quantidade de restrições da formulação, R_f , para as formulações FCA_2 , FCA_3 , FCA_m , e a quantidade de restrições N_{rg} geradas pela formulação $(FCA_1)_t^\pi$ na obtenção de LS .

Para implementar as formulações lineares relaxadas, usamos o software comercial IBM/ILOG CPLEX 12.7 integrado com Ambiente de Desenvolvimento Integrado (IDE - Integrated Development Environment) *Code::Blocks* utilizando a linguagem C++. Os experimentos foram realizados em um processador Intel[®] Core[™] i5 – 4570 com 3.20 GHz, 16 GB RAM e sistema operacional Ubuntu 16.04 LTS.

As tabelas 18 e 19 resumam os resultados obtidos por cada formulação para as instâncias positivas e mistas, respectivamente, onde cada linha, a partir da 3, apresenta o GAP, R_f e N_{rg} para cada par (n, k) . Destacamos em negrito os melhores resultados, conforme descrição dos experimentos. Com relação ao conjunto de instâncias mistas, os resultados mostram que a t -linearização obteve limites mais apertados em comparação a FCA_m , FCA_2 e FCA_3 , em todas as combinações de (n, k) . Já em relação ao conjunto de instâncias com custos positivos, percebe-se uma certa similaridade na qualidade dos limites gerados por $(FCA_1)_t^\pi$ e FCA_m . É importante notar que usamos as restrições fortalecidas em $(FCA_1)_t^\pi$, conforme teoremas 4.1 e 5.1.

A Figura 19 mostra dois gráficos comparando os gaps médios, para as 5 instâncias associadas a um mesmo valor de n , obtidos pelas relaxações FCA_m , FCA_2 , FCA_3 e $(FCA_1)_t^\pi$. O primeiro gráfico refere-se às instâncias positivas e o segundo, às mistas. Eles mostram um comportamento bem parecido dos gaps médios em função de n . É também perceptível no segundo gráfico que a relaxação $(FCA_1)_t^\pi$ obteve em média os melhores limites dentre as relaxações testadas nas instâncias mistas.

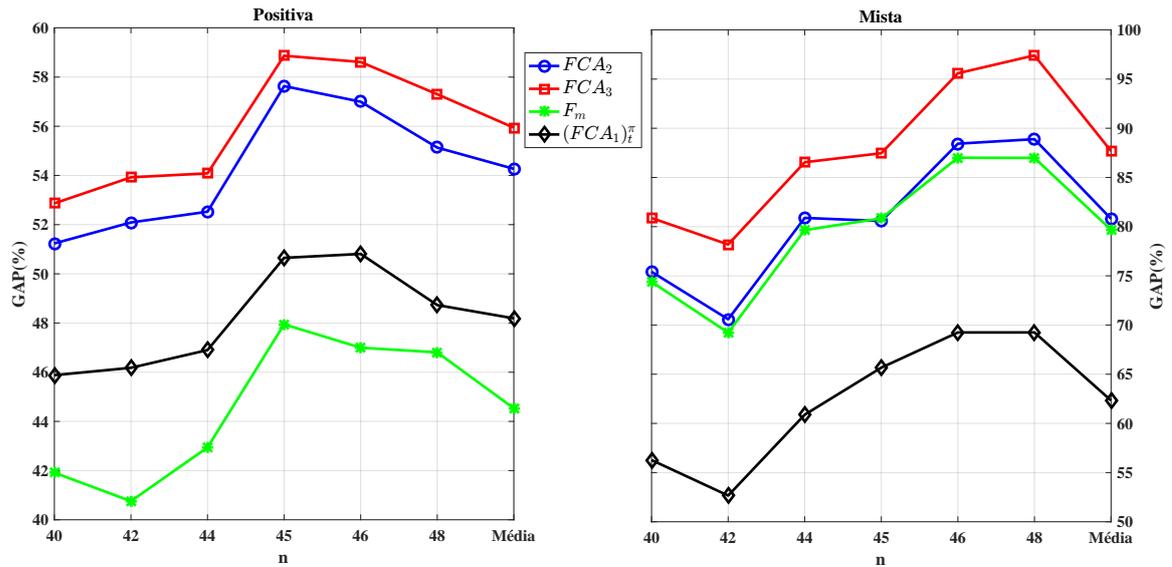
Já a Figura 20 mostra o comparativo entre a quantidade de restrições de $(FCA_1)_t^\pi$ e da formulação FCA_3 , que possui menos restrições dentre as demais. Podemos observar vantagem para $(FCA_1)_t^\pi$ na geração média de restrições das instâncias positivas; por outro lado, nas instâncias mistas, a relaxação $(FCA_1)_t^\pi$ usou quantidade maior de restrições. Esse fato está diretamente relacionado com a melhor qualidade dos limites gerados por $(FCA_1)_t^\pi$.

Tabela 18 – GAP, R_f e N_{rg} obtidos pelas relaxações FCA_m , FCA_2 , FCA_3 e $(FCA_1)_t^\pi$ nas instâncias positivas

n	k	Positiva Valor Ótimo	FCA_m		FCA_2		FCA_3		$(FCA_1)_t^\pi$	
			GAP	R_f	GAP	R_f	GAP	R_f	GAP	N_{rg}
40	1	109346	50.69	1599	71.14	1560	71.28	78	55.69	18
	2	82451	37.21	1599	48.18	1560	49.63	78	42.95	16
	3	68759	43.13	1599	51.55	1560	51.95	78	48.64	15
	4	60782	44.35	1599	48.80	1560	50.52	78	46.72	17
	5	60513	34.17	1599	36.50	1560	40.92	78	35.39	17
42	1	120299	46.12	1763	71.68	1722	72.00	82	56.67	21
	2	87810	44.12	1763	60.03	1722	60.23	82	53.01	18
	3	76554	40.72	1763	50.65	1722	54.05	82	46.97	18
	4	69482	37.05	1763	41.08	1722	43.16	82	39.22	16
	5	67383	35.75	1763	36.99	1722	40.15	82	35.01	20
44	1	136525	45.11	1935	67.70	1892	67.80	86	52.47	22
	2	98186	48.18	1935	59.52	1892	60.42	86	53.69	22
	3	84675	41.06	1935	48.23	1892	51.18	86	44.40	17
	4	75274	42.80	1935	47.29	1892	48.98	86	45.73	15
	5	69540	37.54	1935	39.87	1892	42.02	86	38.21	15
45	1	138694	51.42	2044	74.52	1980	74.96	88	56.79	23
	2	98321	51.19	2044	62.24	1980	63.26	88	55.27	18
	3	82743	44.59	2044	53.25	1980	54.13	88	49.18	17
	4	77500	44.30	2044	48.86	1980	50.71	88	45.64	20
	5	69563	48.20	2044	49.28	1980	51.25	88	46.35	15
46	1	142985	51.91	2135	72.66	2070	73.38	90	57.79	25
	2	108243	47.06	2135	60.80	2070	62.38	90	54.05	22
	3	94859	41.20	2135	50.50	2070	52.99	90	46.28	19
	4	78747	46.97	2135	51.98	2070	52.97	90	49.03	16
	5	72431	47.82	2135	49.03	2070	51.32	90	46.90	16
48	1	163397	49.22	2303	71.38	2256	72.14	94	55.55	24
	2	115471	51.43	2303	62.91	2256	64.49	94	55.87	23
	3	96666	48.39	2303	53.29	2256	55.15	94	49.07	18
	4	88728	43.32	2303	46.16	2256	50.57	94	43.87	20
	5	82117	41.66	2303	41.91	2256	44.14	94	39.31	19
<i>Média Geral</i>			44.56	1963.2	54.26	1913.33	55.94	86.33	48.19	18.73

Fonte: Elaborada pelo autor.

Figura 19: Comparação do GAP médio obtido por FCA_m , FCA_2 , FCA_3 e $(FCA_1)_t^\pi$



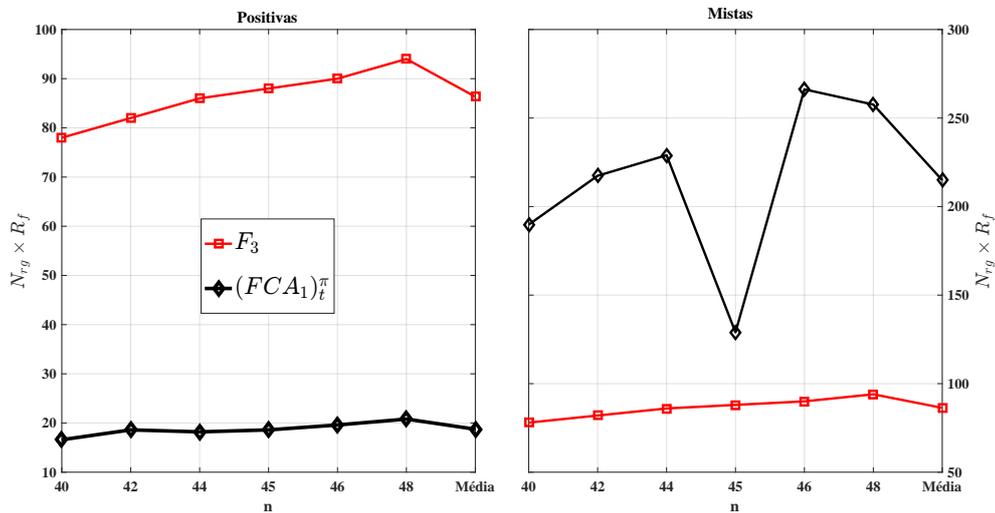
Fonte: Elaborado pelo autor.

Tabela 19 – GAP, R_f e N_{rg} obtidos pelas relaxações FCA_m , FCA_2 , FCA_3 e $(FCA_1)_t^\pi$ nas instâncias mistas

n	k	Mista Valor Ótimo	FCA_m		FCA_2		FCA_3		$(FCA_1)_t^\pi$	
			GAP	R_f	GAP	R_f	GAP	R_f	GAP	N_{rg}
40	1	70348	84.99	2380	90.10	2340	92.02	78	83.11	310
	2	45404	67.21	2380	67.21	2340	73.47	78	52.22	105
	3	34091	81.80	2380	81.80	2340	85.60	78	60.12	175
	4	27758	78.83	2380	78.83	2340	85.58	78	53.59	154
	5	27967	59.09	2380	59.09	2340	67.68	78	32.30	205
42	1	81633	77.68	2625	84.46	2583	87.24	82	74.35	234
	2	46828	87.64	2625	87.64	2583	93.73	82	71.50	224
	3	36689	78.53	2625	78.53	2583	86.86	82	57.52	161
	4	35987	53.45	2625	53.45	2583	61.67	82	32.96	180
	5	35460	48.83	2625	48.83	2583	61.41	82	26.96	288
44	1	90620	77.48	2882	83.42	2838	85.06	86	73.00	328
	2	56960	77.48	2882	77.74	2838	81.51	86	63.90	144
	3	40697	77.95	2882	77.95	2838	90.10	86	56.44	243
	4	32601	86.09	2882	86.09	2838	90.96	86	58.77	229
	5	29407	79.22	2882	79.22	2838	85.08	86	52.29	204
45	1	102295	68.57	3015	77.76	2970	82.15	88	67.88	132
	2	55103	91.78	3015	87.52	2970	92.24	88	75.27	117
	3	43914	69.43	3015	67.37	2970	76.74	88	53.85	115
	4	33990	83.37	3015	81.55	2970	90.80	88	61.72	126
	5	30974	91.04	3015	88.63	2970	95.35	88	69.60	155
46	1	99550	76.07	3151	82.78	3105	84.95	90	73.34	678
	2	58361	88.15	3151	88.58	3105	95.49	90	73.99	185
	3	43915	89.28	3151	89.28	3105	100.15	90	68.46	155
	4	32968	96.21	3151	96.21	3105	101.88	90	71.19	126
	5	31000	85.30	3151	85.30	3105	95.46	90	59.21	187
48	1	113478	76.05	3432	85.17	3384	87.60	94	74.08	549
	2	61768	95.93	3432	96.33	3384	98.39	94	81.57	220
	3	45941	82.52	3432	82.52	3384	93.62	94	62.70	148
	4	36903	85.03	3432	85.03	3384	96.29	94	60.67	206
	5	31351	95.39	3432	95.39	3384	111.10	94	67.19	165
<i>Média Geral</i>			79.68	2941.2	80.79	2870	87.67	86.33	62.33	214.93

Fonte: Elaborada pelo autor.

Figura 20: Comparação do N_{rg} médio obtido por $(FCA_1)_t^\pi$ e FCA_3



Fonte: Elaborado pelo autor.

5.6.2 Um algoritmo branch and bound

A correlação direta entre os resultados sobre fixação de variáveis e desigualdades válidas apresentados na Seção 5.4 e os correspondentes na Seção 4.4 sugere uma adaptação, para o problema PCMPA, do método exato proposto na Subseção 4.6.2 para o PDM, conside-

rando agora coeficientes arbitrários no termo quadrático e m como limite superior para a cardinalidade do conjunto desejado. Uma dificuldade, porém, surge quando pretendemos usar as desigualdades derivadas nas proposições 5.4 e 5.5: as primeiras são válidas apenas para ℓ -soluções; as segundas são inúteis na raiz da árvore, pois tornam-se $\delta_{uv}^\oplus(x, \mathbf{0}) \geq 0$, já que $\delta_{uv}(\mathbf{0}) = 0$. Em outras termos, ambas podem ser efetivas apenas quando usadas em conjunto com o particionamento do problema.

Procurando avaliar uma adaptação mais direta do método exato proposto para o PDM, vamos considerar o caso do PCMPA em que m é exatamente o tamanho do subconjunto desejado. Neste caso, as desigualdades válidas para as m -soluções podem ser usadas no nó raiz. Sendo assim, o método aqui testado é bastante similar àquele da Subseção 4.6.2. No entanto, ele agora engloba os seguintes quatro precedimentos:

- 1– Conversão de (FCA_1) em $(FCA_1)_t^\pi$;
- 2– Separação das restrições de $(FCA_1)_t^\pi$ conforme Teorema 2.5, e fortalecimento da restrição violada segundo Teorema 5.1;
- 3– Obtenção de uma solução viável x_h (limite inferior) através da heurística da Seção 5.5, e aplicação do procedimento de separação em x_h , para gerar a primeira restrição fortalecida;
- 4– Adição à $(FCA_1)_t^\pi$ das desigualdades válidas $\delta_{uv}(x, \bar{x}) \geq M_{uv}(\bar{x})(1 - x_u + x_v)$, para $\bar{x} = \mathbf{0}$, gerando assim o modelo $(BBCA)_t^\pi$:

$$\begin{aligned}
(BBCA)_t^\pi \quad \max \quad & t + \frac{1}{2} \sum_{i \in N} \sum_{j \in N} c_{ji}^- x_i \\
\text{s.a} \quad & t \leq \sum_{\pi(i) \in N_0} \left(s_{\pi(i)}^0 + \sum_{\substack{\pi(j) \in N_0 \\ j < i}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- + \sum_{j > i} -\frac{1}{2} c_{\pi(j)\pi(i)}^- \right) x_{\pi(i)} + \sum_{\pi(i) \in N_0} s_{\pi(i)}^0 \bar{x}_{\pi(i)} \\
& + \sum_{\pi(i) \in N_1} \left(s_{\pi(i)}^1 + \sum_{\substack{\pi(j) \in N_0 \\ j < i}} -\frac{1}{2} c_{\pi(j)\pi(i)}^- \right) x_{\pi(i)} + \sum_{\pi(i) \in N_1} s_{\pi(i)}^1 \bar{x}_{\pi(i)} \quad \forall \pi \in S_n, \forall (N_0, N_1) \in \mathcal{N} \\
& \sum_{\substack{w \in V \setminus \{u, v\} \\ n}} [c(u, w) - c(v, w)] x_w \geq M_{uv}(\mathbf{0})(1 - x_u + x_v) \quad \forall (u, v) \text{ e } \forall (v, u), \\
& \sum_{i=1}^n x_i = m, \quad (x, t) \in \mathbb{B}^n \times \mathbb{R}.
\end{aligned}$$

Na verdade, constatamos através de experimentos que as instâncias mistas aqui utilizadas possuem solução ótima com exatamente m elementos e, portanto, enquadram-se dentro do caso de PCMPA que vamos considerar. Verificamos que, nessas instâncias, embora a quantidade de arestas com pesos negativos seja maior que aquela com pesos positivos, a soma dos pesos destas é maior do que o módulo da soma dos pesos daquelas.

O modelo $(BBCA)_t^\pi$ é resolvido pelo algoritmo branch-and-bound padrão. É importante destacar que o procedimento de separação é usado dentro do branch-and-bound sempre que uma solução inteira é encontrada, ou seja, as restrições da t -linearização

são usadas como *lazy constraints*. Além disso, as restrições t -linearizadas são adicionadas localmente.

Nesta seção, comparamos o desempenho do nosso *branch and bound*, denominado aqui de $(BBCA)_t^\pi$, com a implementação das formulações FCA_1 , FCA_m , FCA_2 e FCA_3 , que foram apresentadas na Seção 5.1. Utilizamos a mesma máquina para melhor comparação entre os métodos: processador Intel[®] Core[™] i5-4570 com 3.20 GHz, 16 GB RAM e sistema operacional Ubuntu 16.04 LTS. As três formulações, assim como o nosso método, foram implementados em linguagem C++ e resolvidos pelo software comercial IBM/ILOG CPLEX 12.7.

Para essa comparação, utilizamos as instâncias positivas e mistas descritas anteriormente, bem como variações dessas instâncias, quando consideramos valores de $m \in \{\lfloor n/4 \rfloor, \lfloor n/8 \rfloor\}$ (além de $m = \lfloor n/2 \rfloor$, como antes). Utilizamos dois parâmetros de medição: o GAP obtido por cada uma das formulações, e o tempo CPU em segundos, sendo esse último parâmetro limitando a 1200 segundos de execução em cada instância. Vale lembrar novamente que, no caso de $(BBCA)_t^\pi$, o CPLEX desabilita operações em paralelo, devido ao uso de *lazy constraints*, diferentemente do que ocorre na resolução das outras formulações.

As tabelas 20 e 21 mostram, a partir da linha 3, o contraste entre o desempenho das formulações para as mesmas instâncias usadas na subseção anterior ($m = \lfloor n/2 \rfloor$), onde destacamos em negrito os melhores resultados para cada instância. Observando o GAP para as instâncias positivas, constatamos que as formulações FCA_1 , FCA_m , FCA_2 , FCA_3 e $(BBCA)_t^\pi$ determinaram o ótimo em, respectivamente, 70% (21 de 30), 100% (30 de 30), 40% (12 de 30), 60% (18 de 30) e 80% (24 de 30) dos casos. Já para as instâncias mistas, os escores são 100% (30 de 30), 100% (30 de 30), 100% (30 de 30), 100% (30 de 30) e 83.3% (25 de 30) para as formulações FCA_m , FCA_2 , FCA_3 e $(BBCA)_t^\pi$, respectivamente. Com relação ao tempo, podemos observar que a formulação FCA_m utiliza em média menor tempo nas instâncias positivas, enquanto que a formulação FCA_1 utiliza em média menor tempo nas instâncias mistas.

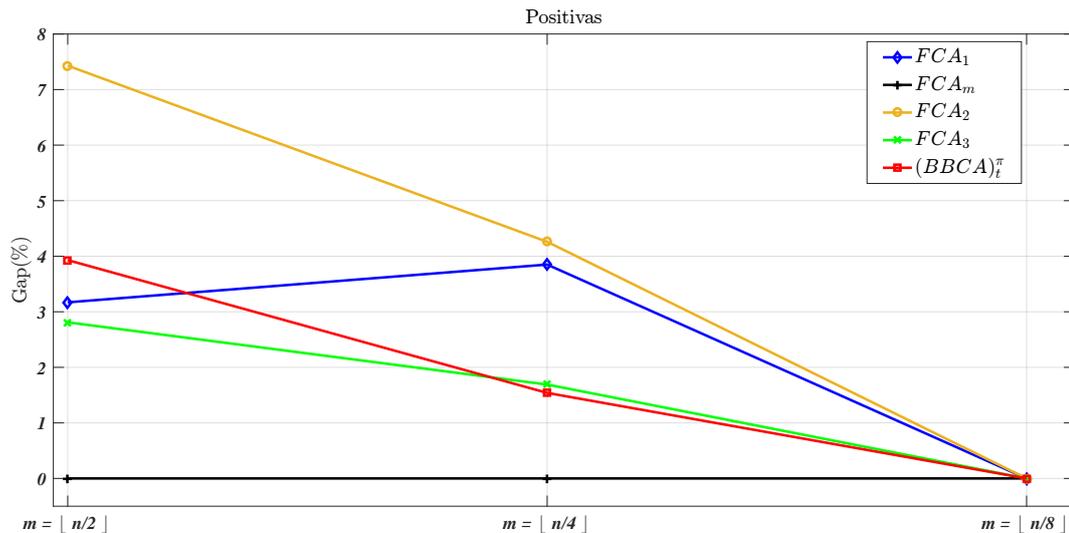
As figuras 21, 22, 23 e 24 mostram as médias de GAP e CPU, quando variamos o valor de $m \in \{\lfloor n/2 \rfloor, \lfloor n/4 \rfloor, \lfloor n/8 \rfloor\}$, obtidas pelas formulações nas instâncias positivas e mistas. Observa-se que todas as formulações, especialmente FCA_1 , FCA_2 , FCA_3 e $(BBCA)_t^\pi$, tendem a obter melhores resultados, em todos os cenários testados, à medida que o valor de m diminui.

Tabela 20: GAP e CPU obtidos pelas formulações FCA_1 , FCA_m , FCA_2 , FCA_3 e $(BBCA)_t^\pi$ nas instâncias positivas com $m = \lfloor n/2 \rfloor$

n	FCA_1		FCA_m		FCA_2		FCA_3		$(BBCA)_t^\pi$	
	GAP	CPU	GAP	CPU	GAP	CPU	GAP	CPU	GAP	CPU
40	4.78	4726.92	0.00	74.78	14.31	1201.55	5.51	1200.01	0.00	434.30
	0.00	149.28	0.00	22.26	0.00	292.78	0.00	34.66	0.00	22.76
	0.00	323.02	0.00	50.53	0.00	538.32	0.00	85.09	0.00	144.90
	0.00	249.63	0.00	57.40	0.00	406.58	0.00	102.68	0.00	99.83
	0.00	34.14	0.00	29.87	0.00	15.26	0.00	5.75	0.00	42.32
42	10.51	4572.22	0.00	116.55	16.85	1201.54	7.52	1200.01	0.00	413.27
	0.00	2338.10	0.00	113.00	10.41	1201.35	1.66	1200.02	0.00	502.02
	0.00	456.28	0.00	63.79	0.00	595.46	0.00	129.12	0.00	135.40
	0.00	65.03	0.00	49.54	0.00	161.37	0.00	28.68	0.00	47.25
	0.00	47.68	0.00	33.08	0.00	126.60	0.00	6.66	0.00	72.78
44	11.97	4586.78	0.00	69.64	17.66	1202.82	7.28	1207.21	0.00	300.74
	0.00	3498.12	0.00	181.27	10.00	1201.31	3.76	1200.02	0.00	465.47
	0.00	218.72	0.00	70.91	0.93	1200.03	0.00	146.63	0.00	278.61
	0.00	326.04	0.00	71.46	0.00	425.58	0.00	214.03	0.00	666.92
	0.00	63.19	0.00	57.00	0.00	243.72	0.00	27.26	0.00	77.50
45	13.59	4603.92	0.00	666.72	21.20	1202.06	10.71	1209.62	0.00	1042.57
	0.00	4547.60	0.00	198.75	14.55	1200.99	5.51	1200.02	0.00	1062.58
	0.00	894.28	0.00	146.69	5.50	1200.02	0.00	683.13	0.00	251.04
	0.00	318.44	0.00	76.95	0.00	722.29	0.00	267.46	0.00	172.81
	0.00	445.35	0.00	70.28	0.00	1030.75	0.00	301.74	0.00	473.44
46	15.91	4610.30	0.00	887.71	22.50	1201.32	11.13	1203.85	24.76	1326.03
	6.46	4698.54	0.00	284.55	16.38	1202.17	6.59	1206.25	19.94	1315.11
	0.00	1980.97	0.00	80.44	4.46	1200.02	0.00	906.42	0.00	827.18
	0.00	1353.29	0.00	163.78	7.73	1201.21	0.00	878.09	12.36	1321.98
	0.00	506.31	0.00	119.54	3.68	1200.03	0.00	485.09	11.91	1319.14
48	15.16	4685.40	0.00	986.43	24.18	1202.73	11.62	1205.07	27.49	1331.88
	10.83	4644.60	0.00	1279.39	20.10	1201.79	9.51	1204.32	21.48	1323.48
	6.06	4664.80	0.00	165.81	9.37	1201.79	3.29	1200.03	0.00	870.62
	0.00	961.74	0.00	123.49	3.15	1200.02	0.00	468.35	0.00	513.07
	0.00	225.77	0.00	63.36	0.00	831.58	0.00	96.40	0.00	701.94
<i>Média</i>	3.17	2026.54	0.00	212.49	7.43	900.43	2.81	643.45	3.93	585.23

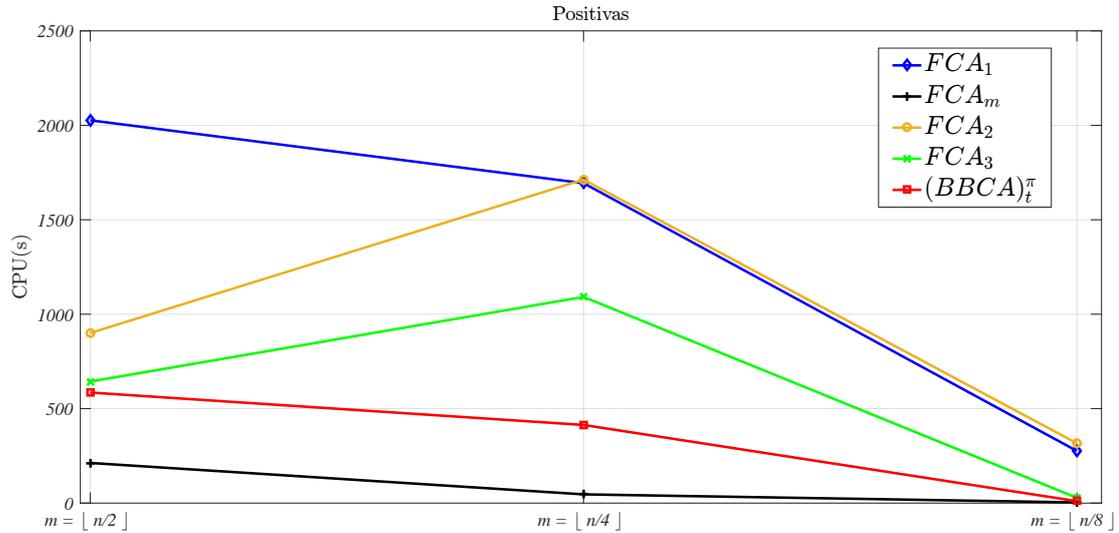
Fonte: Elaborada pelo autor.

Figura 21: Gap médio(30 instâncias positivas) obtido pelas formulações FCA_1 , FCA_m , FCA_2 , FCA_3 e $(BBCA)_t^\pi$



Fonte: Elaborado pelo autor.

Figura 22: Cpu médio(30 instâncias positivas) obtido pelas formulações FCA_1 , FCA_m , FCA_2 , FCA_3 e $(BBCA)_t^\pi$



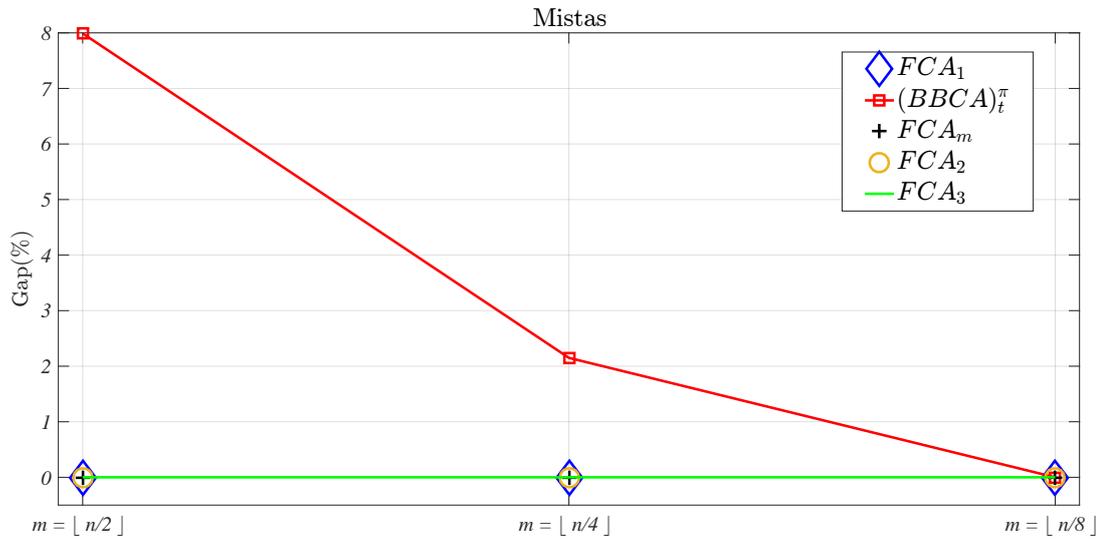
Fonte: Elaborado pelo autor.

Tabela 21: GAP e CPU obtidos pelas formulações FCA_1 , FCA_m , FCA_2 , FCA_3 e $(BBCA)_t^\pi$ nas instâncias mistas com $m = \lfloor n/2 \rfloor$

n	FCA_1		FCA_m		FCA_2		FCA_3		$(BBCA)_t^\pi$	
	GAP	CPU	GAP	CPU	GAP	CPU	GAP	CPU	GAP	CPU
40	0.00	58.71	0.00	91.71	0.00	132.93	0.00	69.96	0.00	154.94
	0.00	13.72	0.00	40.67	0.00	27.60	0.00	3.67	0.00	29.34
	0.00	19.02	0.00	57.58	0.00	31.49	0.00	4.42	0.00	126.45
	0.00	9.98	0.00	40.73	0.00	26.00	0.00	3.42	0.00	43.51
	0.00	2.93	0.00	16.72	0.00	5.61	0.00	3.03	0.00	17.63
42	0.00	113.05	0.00	119.16	0.00	297.87	0.00	100.70	0.00	293.87
	0.00	31.12	0.00	76.87	0.00	65.18	0.00	20.37	0.00	282.70
	0.00	16.10	0.00	52.33	0.00	28.15	0.00	4.53	0.00	66.89
	0.00	5.00	0.00	28.16	0.00	10.52	0.00	1.94	0.00	21.22
	0.00	4.15	0.00	38.25	0.00	11.09	0.00	2.27	0.00	45.97
44	0.00	145.77	0.00	154.49	0.00	485.91	0.00	186.38	0.00	433.76
	0.00	35.37	0.00	92.49	0.00	63.49	0.00	19.88	0.00	317.41
	0.00	21.71	0.00	61.04	0.00	37.49	0.00	7.22	0.00	173.25
	0.00	20.86	0.00	57.21	0.00	40.65	0.00	13.97	0.00	248.21
	0.00	21.13	0.00	62.57	0.00	36.64	0.00	7.24	0.00	122.36
45	0.00	160.34	0.00	107.62	0.00	611.68	0.00	152.98	0.00	327.29
	0.00	46.89	0.00	127.99	0.00	122.13	0.00	56.29	44.33	1322.54
	0.00	19.70	0.00	56.78	0.00	42.24	0.00	3.33	0.00	92.09
	0.00	25.60	0.00	73.20	0.00	39.34	0.00	8.59	0.00	110.27
	0.00	29.88	0.00	83.67	0.00	59.83	0.00	23.02	0.00	352.82
46	0.00	271.30	0.00	142.96	0.00	703.67	0.00	303.52	39.14	1328.13
	0.00	39.23	0.00	149.08	0.00	108.39	0.00	59.31	49.28	1323.86
	0.00	30.34	0.00	102.88	0.00	67.58	0.00	46.33	0.00	348.51
	0.00	34.83	0.00	88.48	0.00	74.83	0.00	26.80	0.00	299.45
	0.00	28.25	0.00	78.59	0.00	47.17	0.00	7.89	0.00	493.84
48	0.00	890.04	0.00	410.62	0.00	1398.41	0.00	1433.98	52.98	1343.18
	0.00	187.55	0.00	305.57	0.00	376.79	0.00	528.58	54.19	1322.48
	0.00	32.38	0.00	87.96	0.00	57.35	0.00	28.00	0.00	514.44
	0.00	26.84	0.00	82.57	0.00	54.64	0.00	13.48	0.00	493.21
	0.00	33.22	0.00	109.57	0.00	68.86	0.00	27.05	0.00	660.72
<i>Média</i>	0.00	79.16	0.00	99.91	0.00	171.11	0.00	105.60	7.99	423.67

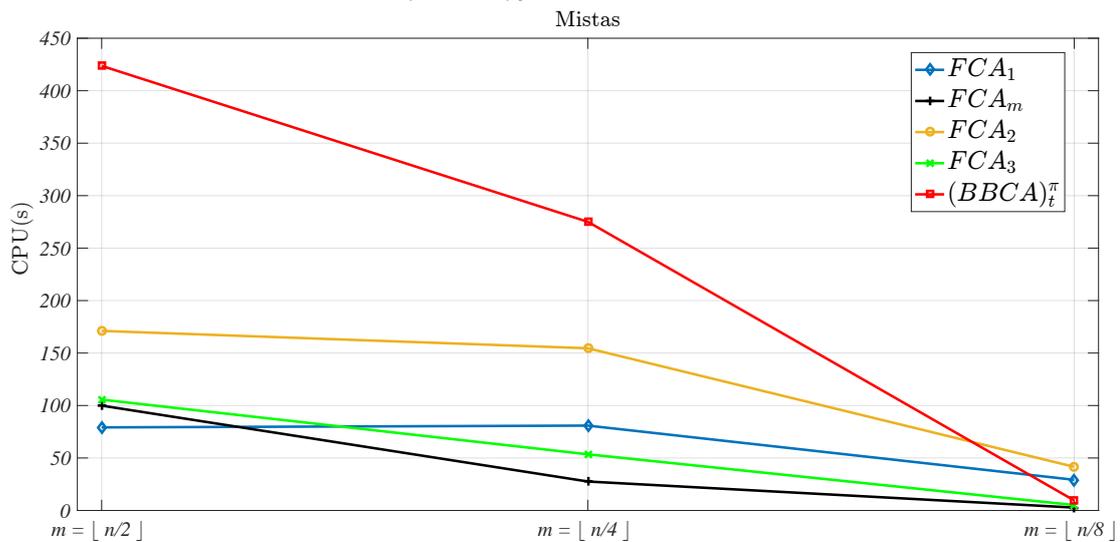
Fonte: Elaborada pelo autor.

Figura 23: Gap médio(30 instâncias mistas) obtido pelas formulações FCA_1 , FCA_m , FCA_2 , FCA_3 e $(BBCA)_t^\pi$



Fonte: Elaborado pelo autor.

Figura 24: Cpu médio(30 instâncias mistas) obtido pelas formulações FCA_1 , FCA_m , FCA_2 , FCA_3 e $(BBCA)_t^\pi$



Fonte: Elaborado pelo autor.

5.7 Conclusão

Neste capítulo, aplicamos a técnica da t -linearização, generalizada no Capítulo 2 para função quadrática com coeficientes arbitrários, ao problema da clique máxima ponderada em aresta (PCMPA). Mostramos como gerar restrições fortalecidas para o PCMPA, quando consideramos a restrição de cardinalidade, estendendo e generalizando a técnica utilizada inicialmente por Rodrigues (2010) no problema da mochila quadrático.

Ficou claro pelos resultados do experimento 1 que a relaxação linear da t -linearização obteve os melhores limites superiores, porém gerando gaps ainda grandes, em

todas as instâncias mistas. Também nas instâncias com custos positivos, essa formulação obteve gap médio comparável ao melhor (de FCA_m), usando o menor número de restrições, entre todas as formulações. Vale mencionar que, para as instâncias mistas, FCA_3 usou o menor número de restrições, porém resultou no maior gap médio.

No experimento 2, comparamos a capacidade das formulações FCA_1 , FCA_m , FCA_2 , FCA_3 e $(BBCA)_t^\pi$ de encontrar a solução ótima para o PCMPA em tempo baixo (limitamos o tempo a 20 minutos). Os resultados mostram que nosso método consegue ser competitivo e até melhor, em comparação com as formulações FCA_1 , FCA_2 e FCA_3 , para as instâncias positivas, à medida que o valor de m diminui. Para as instâncias mistas, o nosso método tende a melhorar à medida que o valor de m diminui, no entanto só se torna competitivo para valores pequenos de m . Por outro lado, a formulação FCA_m obteve vantagem no que diz respeito à quantidade de soluções ótimas alcançadas e tempo computacional requerido nos dois tipos de instâncias e em todas as variações de m .

Há algumas direções de pesquisas futuras, relacionadas a esse contexto. Destacamos duas delas:

1. Efetuar o procedimento de separação nas restrições t -linearizadas já fortalecidas. Atualmente o procedimento de separação encontra a restrição mais violada e em seguida é feito o fortalecimento;
2. Investigar formas eficientes de adicionar as restrições derivadas na Proposição 5.5 na árvore de enumeração, aproveitando seu fortalecimento à medida que variáveis forem fixadas;
3. Implementar um branch and bound ℓ -ário, onde o controle de ramificação seja independente do solver, para investigar o potencial das restrições da Proposição 5.4.

6 CONCLUSÃO

O conteúdo deste trabalho mostra a utilização de diversas técnicas para tratar problemas de otimização, tais como Problema do Corte Máximo(Max-Cut), Problema da Diversidade Máxima(PDM) e Problema da Clique Máxima Ponderada em Arestas(PCMPA), que possuem em sua natureza uma formulação quadrática 0 – 1. Podemos entender esta tese como tendo duas partes principais, que se complementam. A primeira consiste no desenvolvimento teórico de metodologias aplicadas a problemas quadráticos binários quaisquer, assim como a obtenção de propriedades específicas dos problemas citados, algumas das quais podendo aumentar a eficácia dessas metodologias. A segunda parte compreende a proposição de algoritmos exatos e heurísticas, que tomam por base o desenvolvimento teórico realizado na primeira.

Com respeito a parte teórica, pontuamos as principais contribuições. A primeira compreende resultados teóricos relativos à t -linearização, tratada no Capítulo 2. Apresentamos outras demonstrações para resultados de Rodrigues (2010) e Rodrigues *et al.* (2012), mostramos a generalização da técnica para o caso em que os custos do problema são arbitrários e mostramos também como gerar restrições da t -linearização mais fortalecidas, quando o problema possui restrição de cardinalidade.

A segunda contribuição está relacionada ao uso da suavização hiperbólica, abordagem mais comumente empregada a problemas contínuos, também tratada no Capítulo 2. Utilizamos essa técnica no contexto do Max-Cut e mostramos como obter uma versão suavizada do modelo irrestrito definido sobre variáveis indexadas pelos vértices, porém fortalecido pelas restrições triangulares, que são expressas independentemente em função de variáveis indexadas pelas arestas. Comparamos três versões do modelo suavizado, constatando vantagem para a versão que usou as restrições triangulares e um algoritmo de busca local.

A terceira contribuição mostrada nesta tese encontra-se no desenvolvimento de novas restrições válidas para os três problemas aqui trabalhados. Através de uma percepção da função objetivo de cada problema, conseguimos derivar propriedades das soluções ótimas, depois traduzidas em novas restrições válidas. A validade dessa parte do trabalho é corroborada pelos desenvolvimentos teóricos e experimentos computacionais realizados.

A quarta contribuição desta tese consiste na elaboração de heurísticas simples e relativamente rápidas na obtenção de limites. As heurísticas aqui desenvolvidas empregam procedimentos gulosos na construção da solução e, em seguida, efetuam uma busca local, com intuito de alcançar melhores máximos locais.

A segunda parte desta tese trata da avaliação do potencial das técnicas e propriedades aqui desenvolvidas para elaboração de métodos de solução para os problemas estudados. Diversos experimentos computacionais foram planejados e executados com

esse fim. Neles, testamos cada uma das técnicas em separado e comprovamos, através dos resultados, as melhorias obtidas quando usadas. Fizemos a combinação dessas técnicas para construir métodos exatos. Os mesmos foram aplicados ao Max-Cut, PDM e PCMPA. Testes preliminares, que não estão nesta tese, mostram que apesar das melhorias alcançadas na resolução exata do Max-Cut com formulação linearizadas, mais desenvolvimento deve ser feito para que a programação linear se equipare à programação semidefinida no contexto desse problema. Por outro lado, propomos, a nosso conhecimento, um método exato para o PDM capaz de fornecer respostas tão boas e até melhores do que o método de Martí, Gallego, e Duarte (2010) para instâncias da literatura.

6.1 Trabalhos Futuros

Além das direções de pesquisa já apontadas no final dos capítulos, que se apresentam como alternativas ou possíveis melhorias ao que foi aqui proposto, podemos ainda elencar outras possibilidades de linhas de trabalho dentro do contexto apresentado nesta tese. Por exemplo, há outros problemas de dispersão e diversidade que possuem estrutura similar ao PDM e PCMPA, que parecem candidatos naturais à aplicação da t -linearização que apresentamos. Por exemplo, o problema da diversidade máxima de grupos. Esse problema consiste em encontrar uma maneira de particionar um conjunto de n elementos em m subconjuntos menores, chamados grupos, de modo que a diversidade entre os elementos em cada grupo seja a maior possível.

A facilidade que a suavização hiperbólica tem para tratar restrições não lineares abre uma frente de trabalho ampla e ainda pouco explorada em relação à utilização e tratamento de restrições quadráticas em problemas binários. Por exemplo, no problema do Max-Cut, assim como foi feito com as desigualdades triangulares, podemos penalizar e suavizar as restrições apresentadas na Subseção 3.2.3 em sua forma quadrática (36)- (37), sem a necessidade de utilização dos parâmetros M . Imaginamos também a possibilidade de explorar uma restrição de cardinalidade junto com a t -linearização aplicada ao Max-Cut. Note que podemos sempre admitir que uma das partes do corte tem cardinalidade no máximo $\lfloor n/2 \rfloor$. Podemos considerar estratégias para particionar o problema usando diferentes limites para tamanhos de uma parte e poder derivar restrições t -linearizadas mais fortes.

REFERÊNCIAS

- Alidaee, Bahram; Glover, Fred; Kochenberger, Gary; Wang, Haibo. Solving the maximum edge weight clique problem via unconstrained quadratic programming. *European Journal of Operational Research*, v. 181, n. 2, p. 592–597, 2007.
- Alperin, H.; Nowak, I. Lagrangian Smoothing Heuristic for Max-Cut. *Journal of Heuristic*, v. 11, n. 5–6, p. 447–463, 2005.
- Anjos, Miguel F.; Lasserre, Jean B. Introduction to semidefinite, conic and polynomial optimization. *Handbook on semidefinite, conic and polynomial optimization*, Springer, v. 166, p. 1–22. 2012.
- Aringhieri, Roberto; Cordone, Roberto. Comparing local search metaheuristics for the maximum diversity problem. *Journal of the Operational research Society*, v. 62, n. 2, p. 266–280, 2011.
- Bagirov, A. M.; Al Nuaimat, A.; Sultanova, N. Hyperbolic smoothing function method for minimax problems. *Optimization*, v. 62, n. 6, p. 759–782, 2013.
- Bagirov, Adil M.; Ordin, Burak; Ozturk, Gurkan; Xavier, Adilson E. An incremental clustering algorithm based on hyperbolic smoothing. *Computational Optimization and Applications*, v. 61, n. 1, p. 219–241, 2015.
- Balas, Egon. Extension de l’algorithme additif à la programmation en nombres entiers et à la programmation non linéaire. *Comptes Rendus Hebdomadaires des Seances de L’Academie des Sciences*, v. 258, n. 21, p. 5136, 1964.
- Barahona, Francisco. The max-cut problem on graphs not contractible to K_5 . *Operations Research Letters*, v. 2, n. 3, p. 107–111, 1983.
- Barahona, Francisco; Grötschel, Martin; Jünger, Michael; Reinelt, Gerhard. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, v. 36, n. 3, p. 493–513, 1988.
- Barahona, Francisco; Mahjoub, Ali Ridha. On the cut polytope. *Mathematical programming*, v. 36, n. 2, p. 157–173, 1986.
- Ben-Ameur, Walid; Mahjoub, Ali Ridha; Neto, José. The maximum cut problem. *Paradigms of Combinatorial Optimization: Problems and New Approaches, Volume 2*, p. 131–172, 2006.
- Benlic, Una; Hao, Jin-Kao. Breakout local search for the max-cut problem. *Engineering Applications of Artificial Intelligence*, v. 26, n. 3, p. 1162–1173, 2013.

- Benson, Steven J; Ye, Yinyu; Zhang, Xiong. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, v. 10, n. 2, p. 443–461, 2000.
- Besanko, David; Perry, Martin K.; Spady, Richard H. The logit model of monopolistic competition: brand diversity. *The Journal of Industrial Economics*, p. 397–415, 1990.
- Billionnet, Alain. Different formulations for solving the heaviest k-subgraph problem. *INFOR: Information Systems and Operational Research*, v. 43, n. 3, p. 171–186, 2005.
- Billionnet, Alain; Sutter, Alain. Minimization of a quadratic pseudo-boolean function. *European Journal of Operational Research*, v. 78, n. 1, p. 106–115, 1994.
- Bonato, Thorsten; Jünger, Michael; Reinelt, Gerhard; Rinaldi, Giovanni. Lifting and separation procedures for the cut polytope. *Mathematical Programming*, v. 146, n. 1-2, p. 351–378, 2014.
- Bonomo, Flavia; Marenco, Javier; Sabán, Daniela; Stier-Moses, Nicolás. A polyhedral study of the maximum edge subgraph problem. *Electronic Notes in Discrete Mathematics*, v. 35, p. 197–202, 2009.
- Boros, Endre; Hammer, Peter L. The max-cut problem and quadratic 0–1 optimization; polyhedral aspects, relaxations and bounds. *Annals of Operations Research*, v. 33, n. 3, p. 151–180, 1991.
- Boros, Endre; Hammer, Peter L. Pseudo-boolean optimization. *Discrete Applied Mathematics*, v. 123, n. 1, p. 155–225, 2002.
- Burer, Samuel; Monteiro, Renato D. C.; Zhang, Yin. Rank-two relaxation heuristics for max-cut and other binary quadratic programs. *SIAM Journal on Optimization*, v. 12, n. 2, p. 503–521, 2002.
- Chardaire, Pierre; Sutter, Alain. A decomposition method for quadratic zero-one programming. *Management Science*, v. 41, n. 4, p. 704–712, 1995.
- Church, Richard; Revelle, Charles. The maximal covering location problem. *Papers of the Regional Science Association*, v. 32, n. 1, p. 101–118, 1974.
- Dai, Yu-Hong. A perfect example for the BFGS method. *Mathematical Programming*, v. 138, n. 1-2, p. 501–530, 2013.
- Dantzig, George B. Linear programming and extensions. 1963.
- Deza, Michel Marie; Laurent, Monique. *Geometry of cuts e metrics*, v. 15. Springer, 2009.

Dijkhuizen, G.; Faigle, U. A cutting-plane approach to the edge-weighted maximal clique problem. *European Journal of Operational Research*, v. 69, n. 1, p. 121 – 130, 1993.

Elloumi, Sourour; Faye, Alain; Soutif, Eric. Decomposition and linearization for 0-1 quadratic programming. *Annals of Operations Research*, v. 99, n. 1, p. 79–93, 2000.

Erkut, Erhan. The discrete p-dispersion problem. *European Journal of Operational Research*, v. 46, n. 1, p. 48–60, 1990.

Erkut, Erhan; Neuman, Susan. Comparison of four models for dispersing facilities. *INFOR: Information Systems and Operational Research*, v. 29, n. 2, p. 68–86, 1991.

Feige, Uriel; Peleg, David; Kortsarz, Guy. The dense k-subgraph problem. *Algorithmica*, v. 29, n. 3, p. 410–421, 2001.

Ferreira, Carlos Eduardo; Fernandes, C.; Miyazawa, Flávio; Soares, J. A. R.; Pina Jr, J. C.; Guimarães, K. S.; Carvalho, M. H.; Cerioli, M. R.; Feofiloff, Paulo; Dahab, R.; Wakabayashi, Y. Uma introdução sucinta a algoritmos de aproximação. *Colóquio Brasileiro de Matemática-IMPA, Rio de Janeiro-RJ*, 2001.

Festa, Paola; Pardalos, Panos M.; Resende, Mauricio G. C.; Ribeiro, Celso C. Randomized heuristic for the max-cut problem. *Siam Journal on Optimization*, v. 7, p. 1033–1058, 2002.

Fischer, Ilse; Gruber, Gerald; Rendl, Franz; Sotirov, Renata. Computational experience with a bundle approach for semidefinite cutting plane relaxations of max-cut and equipartition. *Mathematical Programming Series B*, v. 105, n. 2–3, p. 451–469, 2006.

Fortet, Robert. Applications de l’algèbre de boole en recherché opérationnelle. *Revue française d’automatique d’informatique et de recherche opérationnelle*, v. 4, p. 5–36, 1959a.

Fortet, Robert. L’algebre de boole et ses applications en recherche operationnelle. *Cahiers du centre d’etudes de recherche opérationnelle*, v. 4, p. 5–36, 1959b.

Gary, Michael R.; Johnson, David S. Computers and intractability: a guide to the theory of NP-completeness. 1979.

Ghyczy, T. Von. Product diversity e proliferation as a new mode of competing in the motor car. *International Journal of Vehicle Design*, v. 6, n. 4-5, p. 423–425, 1985.

Giffler, Bernard; Thompson, Gerald Luther. Algorithms for solving production-scheduling problems. *Operations research*, v. 8, n. 4, p. 487–503, 1960.

Glover, Fred. Improved linear integer programming formulations of nonlinear integer

problems. *Management Science*, v. 22, n. 4, p. 455–460, 1975.

Glover, Fred; Kuo, Ching-Chung; Dhir, Krishna S. Heuristic algorithms for the maximum diversity problem. *Journal of information and Optimization Sciences*, v. 19, n. 1, p. 109–132, 1998.

Glover, Fred; Woolsey, Eugene. Converting the 0-1 polynomial programming problem to a 0-1 linear program. *Operations research*, v. 22, n. 1, p. 180–182, 1974.

Goemans, Michel X.; Williamson, David P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, v. 42, n. 6, p. 1115–1145, 1995.

Grötschel, M.; Wakabayashi, Y. A Cutting Plane Algorithm for a Clustering Problem. *Math. Program.*, v. 45, n. 1, p. 59–96, 1989.

Gueye, Serigne; Michelon, Philippe. A linearization framework for unconstrained quadratic (0-1) problems. *Discrete Applied Mathematics*, v. 157, n. 6, p. 1255–1266, 2009.

Hadlock, Frank. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, v. 4, n. 3, p. 221–225, 1975.

Hammer, Peter L.; Rudeanu, Sergiu. *Boolean methods in operations research*. Springer-Verlag, 1968.

Hansen, Pierre. Methods of nonlinear 0-1 programming. *Annals of Discrete Mathematics*, v. 5, p. 53–70, 1979.

Helmberg, Christoph; Rendl, Franz; Vanderbei, Robert J.; Wolkowicz, Henry. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, v. 6, n. 2, p. 342–361, 1996.

Hu, T. C. *Integer programming and network flows*. Tech. rep., DTIC Document, 1969.

Karp, Richard M. *Reducibility among combinatorial problems*. Springer, 1972.

Kececioglu, John D.; Myers, Eugene W. Combinatorial algorithms for dna sequence assembly. *Algorithmica*, v. 13, n. 1-2, p. 7, 1995.

Krislock, N.; Malick, J.; Rouoin, F. Improved semidefinite bounding procedure for solving max-cut problems to optimality. *Mathematical Programming*, v. 143, n. 1–2, p. 61–86, 2014.

Kuo, Ching-Chung; Glover, Fred; Dhir, Krishna S. Analyzing and modeling the

maximum diversity problem by zero-one programming. *Decision Sciences*, v. 24, n. 6, p. 1171–1185, 1993.

Lavor, Carlile. *Suavização hiperbólica aplicada no problema de localização de estações de rádio base*. Universidade Federal do Rio de Janeiro, Rio de Janeiro, The address of the publisher, 2004. An optional note.

Loiola, Eliane Maria; de Abreu, Nair Maria Maia; Boaventura-Netto, Paulo Oswaldo; Hahn, Peter; Querido, Tania. A survey for the quadratic assignment problem. *European journal of operational research*, v. 176, n. 2, p. 657–690, 2007.

Macambira, Elder Magalhães. An application of tabu search heuristic for the maximum edge-weighted subgraph problem. *Annals of Operations Research*, v. 117, n. 1-4, p. 175–190, 2002.

Macambira, Elder Magalhães; Souza, Cid Carvalho de. The edge-weighted clique problem: valid inequalities, facets and polyhedral computations. *European Journal of Operational Research*, v. 123, n. 2, p. 346–371, 2000.

Martí, Rafael; Duarte, Abraham; Laguna, Manuel. Advanced scatter search for the max-cut problem. *INFORMS Journal on Computing*, v. 21, n. 1, p. 26–38, 2009.

Martí, Rafael; Gallego, Micael; Duarte, Abraham. A branch and bound algorithm for the maximum diversity problem. *European Journal of Operational Research*, v. 200, n. 1, p. 36–44, 2010.

Martí, Rafael; Gallego, Micael; Duarte, Abraham; Pardo, Eduardo G. Heuristics and metaheuristics for the maximum diversity problem. *Journal of Heuristics*, v. 19, n. 4, p. 591–615, 2013.

McConnell, Scott. The new battle over immigration. *Fortune*, v. 117, n. 10, p. 89, 1988.

Miyazawa, Flávio K.; de Souza, Cid C. Introdução à otimização combinatória. *Jornadas de Atualização em Informática-Congresso da Sociedade Brasileira de Computação-JAI-SBC*. 2015.

O'Kelly, Morton E. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, v. 32, n. 3, p. 393–404, 1987.

Orlova, G. I. Finding maximum cut in a graph. *Engineering Cybernetics*, v. 10, n. 3, p. 502–506, 1972.

Palagi, Laura; Piccialli, Veronica; Rendl, Franz; Rinaldi, Giovanni; Wiegele, Angelika. Computational approaches to max-cut. *Handbook on semidefinite, conic and polynomial optimization*, Springer, p. 821–847. 2012.

Pearce, David. Economics e genetic diversity. *Futures*, v. 19, n. 6, p. 710–712, 1987.

Pisinger, David. Upper bounds and exact algorithms for p-dispersion problems. *Computers & operations research*, v. 33, n. 5, p. 1380–1398, 2006.

Porter, W. M.; Rawal, K. M.; Rachie, K. O.; Wien, H. C.; Williams, R. C. Cowpea germplasm catalog No 1. *International institute of tropical agriculture, ibadan, nigeria*, 1975.

Ravi, Sekharipuram S.; Rosenkrantz, Daniel J.; Tayi, Giri Kumar. Heuristic and special case algorithms for dispersion problems. *Operations Research*, v. 42, n. 2, p. 299–310, 1994.

Rendl, Franz; Rinaldi, Giovanni; Wiegele, Angelika. A branch and bound algorithm for max-cut based on combining semidefinite and polyhedral relaxations. *International Conference on Integer Programming and Combinatorial Optimization*. Springer, 2007, p. 295–309.

Rendl, Franz; Rinaldi, Giovanni; Wiegele, Angelika. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming*, v. 121, n. 2, p. 307–335, 2010.

Rinaldi, Giovanni. Rudy. 1998. URL
<http://www-user.tu-chemnitz.de/~helmbert/rudy.tar.gz>.

Rodrigues, Carlos; Quadri, Dominique; Michelon, Philippe; Gueye, Serigne. 0-1 quadratic knapsack problems: an exact approach based on a t-linearization. *SIAM Journal on Optimization*, v. 22, n. 4, p. 1449–1468, 2012.

Rodrigues, Carlos Diego. *Abordagens híbridadas na solução de problemas de programação inteira da teoria e prática*. Universidade Federal do Ceará e Université d'Avignon, The address of the publisher, 2010. An optional note.

Santos, A. B. A. *Problemas de programação não-diferenciável: Uma metodologia de suavização*. M. Sc. Thesis, Federal University of Rio de Janeiro, Rio de Janeiro, The address of the publisher, 1997. An optional note.

Shylo, V. P.; Shylo, O. V.; Roschyn, V. A. Solving weighted max-cut problem by global equilibrium search. *Cybernetics and Systems Analysis*, v. 48, n. 4, p. 563–567, 2012.

Silva, Geiza C.; Ochi, Simone L, Luiz S. e Martins. Experimental comparison of greedy randomized adaptive search procedures for the maximum diversity problem. *International Workshop on Experimental and Efficient Algorithms*. Springer, 2004, p. 498–512.

Soares, Pablo; Campêlo, Manoel. t-Linearização aplicada ao problema da clique máxima

ponderada em arestas. *Anais do L SBPO*. 2018.

Soares, Pablo; Campêlo, Manoel; Rebouças, Daniel. Método exato para o problema da diversidade máxima. *Anais do XIX Claio*. 2018.

Soares, Pablo; Campêlo, Manoel; Rodrigues, Carlos Diego; Michelon, Philippe. t-Linearização de funções quadráticas de variáveis binárias. *Anais do XLIX SBPO*. 2017, p. 2569–2580.

Sørensen, Michael M. New facets and a branch-and-cut algorithm for the weighted clique problem. *European Journal of Operational Research*, v. 154, n. 1, p. 57–70, 2004.

Souza, Michael; Xavier, Adilson Elias; Lavor, Carlile; Maculan, Nelson. Hyperbolic smoothing and penalty techniques applied to molecular structure determination. *Operations Research Letters*, v. 39, n. 6, p. 461–465, 2011.

Swierenga, Robert P. Ethnicity in historical perspective. *Social Science*, p. 31–44, 1977.

Toth, Paolo; Vigo, Daniele. *Vehicle routing: problems, methods, and applications*. SIAM, 2014.

Ullman, Jeffrey D. NP-complete scheduling problems. *Journal of Computer and System sciences*, v. 10, n. 3, p. 384–393, 1975.

Vandenberghe, Lieven; Boyd, Stephen. Semidefinite programming. *SIAM review*, v. 38, n. 1, p. 49–95, 1996.

Wang, Yang; Lü, Zhipeng; Glover, Fred; Hao, Jin-Kao. Path relinking for unconstrained binary quadratic programming. *European Journal of Operational Research*, v. 223, n. 3, p. 595–604, 2012.

Warner, D. Michael; Prawda, Juan. A mathematical programming model for scheduling nursing personnel in a hospital. *Management Science*, v. 19, n. 4-part-1, p. 411–422, 1972.

Wiegele, Angelika. Biq Mac Library—A collection of Max-Cut and quadratic 0-1 programming instances of medium size. *Preprint*, 2007.

Wolsey, Laurence A. *Integer Programming*. Wiley, 1. ed., 1998.

Wu, Qinghua; Wang, Yang; Lü, Zhipeng. A tabu search based hybrid evolutionary algorithm for the max-cut problem. *Applied Soft Computing*, v. 34, p. 827–837, 2015.

Xavier, Adilson Elias. *Penalização hiperbólica: um novo método para resolução de problemas de otimização*. M. Sc. Thesis-COPPE-UFRJ, Rio de Janeiro, The address of

the publisher, 1982. An optional note.

Xavier, Adilson Elias. The hyperbolic smoothing clustering method. *Pattern Recognition*, v. 43, n. 3, p. 731–737, 2010.

Xavier, Adilson Elias; Xavier, Vinicius Layter. Solving the minimum sum-of-squares clustering problem by hyperbolic smoothing and partition into boundary and gravitational regions. *Pattern Recognition*, v. 44, n. 1, p. 70–77, 2011.

Xavier, Adilson Elias; Xavier, Vinicius Layter. Flying elephants: a general method for solving non-differentiable problems. *Journal of Heuristics*, v. 22, n. 4, p. 649–664, 2016.

Xavier, Vinicius L.; França, Felipe M. G.; Xavier, Adilson E.; Lima, Priscila M. V. A hyperbolic smoothing approach to the multisource weber problem. *Journal of Global Optimization*, v. 60, n. 1, p. 49–58, 2014.

Xia, Y.; Xu, Z. An efficient lagrangian smooting heuristic for max-Cut. *Indian Journal of Pure and Applied Mathematics*, v. 41, n. 5, p. 683–700, 2010.

Zhou, Yangming; Hao, Jin-Kao; Duval, Béatrice. Opposition-based memetic search for the maximum diversity problem. *IEEE Transactions on Evolutionary Computation*, v. 21, n. 5, p. 731–745, 2017.